

# **”Olen uskaltanut aloittaa ohjelmoinnin opetuksen”**

## **Ohjelmoinnin täydennyskoulutuksen vaikuttavuus luokanopettajien kokemana**

Helsingin yliopisto  
Kasvatustieteellinen tiedekunta  
Opettajankoulutuslaitos  
Luokanopettajan koulutus  
Pro gradu -tutkielma  
Kasvatustiede  
Huhtikuu 2019  
Julia Luhtala

Ohjaajat: Tiina Korhonen, Anu  
Laine ja Markku Hannula



Tiedekunta - Fakultet - Faculty Kasvatustieteellinen tiedekunta		Laitos - Institution - Department Opettajankoulutuslaitos	
Tekijä - Författare - Author Julia Luhtala			
Työn nimi - Arbetets titel "Olen uskaltanut aloittaa ohjelmoinnin opetuksen" – Ohjelmoinnin täydennyskoulutuksen vaikuttavuus luokanopettajien kokemana			
Title "I have dared to begin to teach programming" – Class teachers thoughts of continuing educations effects to teach programming			
Oppiaine - Läroämne - Subject Kasvatustiede			
Työn laji/ Ohjaaja - Arbetets art/Handledare - Level/Instructor Pro gradu -tutkielma / Tiina Korhonen, Anu Laine ja Markku Hannula		Aika - Datum - Month and year Huhtikuu 2019	Sivumäärä - Sidoantal - Number of pages 62 s
Tiivistelmä - Referat - Abstract <p><i>Tavoitteet.</i> Ohjelmointiin liittyvää ymmärrystä pidetään tärkeänä osana teknologista lukutaitoa. Ohjelmoinnillista, eli algoritmista, ajattelua pidetään perustavanlaatuisena taitona meille kaikille. Muun muassa tämän vuoksi ohjelmointi tuli osaksi opetussuunnitelmaa syksyllä 2016. Samalla ohjelmointiin liittyvän täydennyskoulutuksen tarve kasvoi. Tämän tutkimuksen tavoitteena on analysoida ja tulkita täydennyskoulutuksen vaikutuksia luokanopettajien ohjelmoinnin opetukseen. Lisäksi tarkoituksenani on selvittää, millä tavalla luokanopettajat itse ajattelevat toteuttavansa ohjelmoinnin opetusta ennen täydennyskoulutusta, sen jälkeen sekä tulevaisuudessa.</p> <p><i>Menetelmät.</i> Aineisto koostui 101 luokanopettajan kyselylomakevastauksista koskien ohjelmoinnin opettamista. Kaikki tutkittavat olivat käyneet vähintään yhden ohjelmointiin liittyvän täydennyskoulutuksen. Tutkittavat valikoituivat laajemmasta aineistosta sen perusteella, että kaikki heistä olivat vastanneet kahteen eri kyselyyn, mikä mahdollisti muutoksen tutkimisen. Tämän lisäksi vastaajia yhdisti se, että kaikki opettivat 3.–6. luokkia. Aineisto analysoitiin aineistolähtöistä sisällönanalyysia hyödyntäen.</p> <p><i>Tulokset ja johtopäätökset.</i> Tutkimuksessa selvisi, että luokanopettajat toteuttivat ohjelmoinnin opetusta enemmän ja monipuolisemmin täydennyskoulutuksen jälkeen. Luokanopettajat myös raportoivat hyödyntäneensä erilaisia ohjelmointiympäristöjä, robotteja sekä yhteistyötä kollegan kanssa ennen täydennyskoulutusta, sen jälkeen ja myös tulevaisuudessa. Täydennyskoulutuksen positiiviset vaikutukset nousivat esille huomattavasti useammin kuin negatiiviset vaikutukset. Täydennyskoulutuksen vaikutuksista opettajat raportoivat tietojensa ja taitojensa karttumisesta, oman työn helpottumisesta sekä innostuksen lisääntymisestä. Tuloksia voi hyödyntää täydennyskoulutuksen hyötyjä perustellessa. Tulokset voidaan myös nähdä voimavarana ohjelmoinnin täydennyskoulutuksien kehittämisessä.</p>			
Avainsanat - Nyckelord Ohjelmointi, täydennyskoulutus			
Keywords Programming, computing, continuing education			
Säilytyspaikka - Förvaringsställe - Where deposited Helsingin yliopiston kirjasto – Helda / E-thesis (opinnäytteet) <a href="https://ethesis.helsinki.fi">ethesis.helsinki.fi</a>			
Muita tietoja - Övriga uppgifter - Additional information			



Tiedekunta - Fakultet - Faculty Educational Sciences		Laitos - Institution - Department Teacher Education	
Tekijä - Författare - Author Julia Luhtala			
Työn nimi - Arbetets titel "Olen uskaltanut aloittaa ohjelmoinnin opetuksen" – Ohjelmoinnin täydennyskoulutuksen vaikuttavuus luokanopettajien kokemana			
Title "I have dared to start to teach programming" – Class teachers' thoughts of continuing education effects on teaching programming			
Oppiaine - Läroämne - Subject Education			
Työn laji/ Ohjaaja - Arbetets art/Handledare - Level/Instructor Master's Thesis / Tiina Korhonen, Anu Laine, and Markku Hannula		Aika - Datum - Month and year April 2019	Sivumäärä - Sidoantal - Number of pages 62 pp.
Tiivistelmä - Referat - Abstract <p><i>Objectives.</i> Understanding programming is considered an important part of technological literacy. Computational or algorithmic thinking is regarded as a fundamental skill for everyone. These are some of the reasons why programming became part of the curriculum in autumn 2016. This also increased the need for further education in programming. The purpose of this thesis is to analyse and interpret how the continuing education affects the programming teaching of class teachers. In addition, I also examine how the class teachers themselves think they will carry out programming teaching before continuing education course, after it and in the future.</p> <p><i>Methods.</i> The material consisted of questionnaire replies from 101 class teachers on how to teach programming. All the subjects had completed at least one programming related continuing education course. The subjects were selected out of a broader data set based on that all of them had responded to two different surveys, which allowed examining the change. In addition, respondents were united by the fact that everyone taught grades 3–6. The material was analysed using data based content analysis.</p> <p><i>Results and conclusions.</i> The study revealed that class teachers carried out more programming lessons which were also more versatile after participating in the continuing education. The class teachers also reported using different learning environments, robots and cooperation with their colleagues before and after the continuing education and in the future. The positive effects of the continuing education were mentioned much more often than the negative effects by class teachers. Some of these effects the teachers reported were increased knowledge and skills, increased enthusiasm and that working felt easier. These results can be used in arguing the usefulness of continuing education. In addition, the results can be seen as a resource in developing continuing education of programming.</p>			
Avainsanat - Nyckelord Ohjelmointi, täydennyskoulutus			
Keywords Programming, computing, continuing education			
Säilytyspaikka - Förvaringsställe - Where deposited University of Helsinki, City Centre Campus Library / Helda / E-thesis, <a href="https://ethesis.helsinki.fi">ethesis.helsinki.fi</a>			
Muita tietoja - Övriga uppgifter - Additional information			

# Sisällys

1	JOHDANTO .....	2
2	OHJELMOINTI PERUSKOULUSSA .....	5
2.1	Algoritminen ajattelu .....	5
2.2	Ohjelmointi opetussuunnitelmassa .....	8
2.3	Ohjelmoinnin opetus .....	10
2.4	Yhteenveto.....	15
3	TÄYDENNYSKOULUTUS.....	16
3.1	Tarve ja muodot .....	16
3.2	Vaikuttavuus .....	17
4	TUTKIMUSTEHTÄVÄ JA TUTKIMUSKYSYMYKSET .....	20
5	TUTKIMUKSEN TOTEUTUS .....	21
5.1	Tutkimusasetelma.....	21
5.2	Aineisto .....	22
5.3	Aineiston analyysi .....	22
6	TUTKIMUSTULOKSET .....	25
6.1	Täydennyskoulutuksen vaikutukset ohjelmoinnin opettamiseen.....	25
6.1.1	Täydennyskoulutusta ennen .....	25
6.1.2	Täydennyskoulutuksen jälkeen .....	31
6.1.3	Tulevaisuudessa .....	37
6.1.4	Yhteenveto.....	43
6.2	Luokanopettajien ajatuksia täydennyskoulutuksen vaikutuksista ohjelmoinnin opettamiseen .....	46
7	LUOTETTAVUUS .....	53
8	POHDINTA .....	55
	LÄHTEET .....	59

## Taulukot

Taulukko 1: Ohjelmointi opetussuunnitelmassa (POPS 2014).....	9
Taulukko 2: Esimerkki vastauksen pelkistämisestä sekä ylä- ja alaluokkien muodostamisesta.....	23

## Kuviot

Kuvio 1: Algoritminen ajattelu (Selby & Woollard, 2014; Liukas & Mykkänen, 2014, 76–78).....	6
Kuvio 2: Algoritminen ajattelu Futschekin (2006, 160) mukaan.....	7
Kuvio 3: Vastaajien ( $N = 101$ ) arviot siitä, kuinka paljon he olivat opettaneet ohjelmointia ennen osallistumista täydennyskoulutukseen. ....	26
Kuvio 4: Yhteenveto kaikkien ennen täydennyskoulutusta ohjelmointia opettaneiden vastauksista.....	30
Kuvio 5: Ei lainkaan ennen täydennyskoulutusta ohjelmointia opettaneiden vastaukset koulutuksen jälkeen toteuttamastaan opetuksesta.....	31
Kuvio 6: Yhteenveto kaikkien opettajien vastauksista koskien täydennyskoulutuksen jälkeen toteutettua ohjelmoinnin opetusta.....	37
Kuvio 7: Maininnat koskien opettajien vastauksia ohjelmoinnin opettamisesta tulevaisuudessa. ....	38
Kuvio 8: Laitteita ja ohjelmointiympäristöjä koskevat maininnat opettajien vastauksissa koskien ohjelmoinnin opettamista tulevaisuudessa. ....	39
Kuvio 9: Toimintatapoja koskevat maininnat opettajien vastauksissa koskien ohjelmoinnin opettamista tulevaisuudessa.....	40
Kuvio 10: Yhteenveto kaikkien opettajien vastauksista koskien tulevaisuudessa toteutettavaa ohjelmoinnin opetusta. ....	43
Kuvio 11: Opettajien ajatukset ohjelmoinnin opettamisesta ennen täydennyskoulutusta, sen jälkeen sekä tulevaisuudessa. ....	44
Kuvio 12: Opettajien vastaukset siitä, millä tavalla koulutus vaikutti heidän ohjelmoinnin opetukseensa jaoteltuina positiivisiin, negatiivisiin, tyhjiin sekä muihin vastauksiin.....	46
Kuvio 13: Täydennyskoulutuksen positiiviset vaikutukset luokittain. ....	47

# 1 Johdanto

Alati muuttuvassa yhteiskunnassa, globaalissa maailmassa, tarvittavat taidot sekä osaaminen muuttuvat kovalla tahdilla. Jo 1980-luvulla jotkut asiantuntijat näkivät tietotekniikan oleellisena tulevaisuuden perustaitona (Saarikoski, 2004, 92–93). Myöhemmin, vuonna 2011, Salo, Kankaanranta, Vähähyyppä ja Viik-Kajander selvittivät eri alojen asiantuntijoiden näkemyksiä vuonna 2020 tarvittavista taidoista ja osaamisesta. Tällöin esiin nousivat muun muassa tietotulvan hallitseminen ja siinä käytettävät tietotekniset apuvälineet. Oppimista edistävien tietoteknisten apuvälineiden lisäksi esille nousivat mm. kannettavat päätelaitteet, pelit ja simulaatiot, verkko-opiskelu sekä erilaiset tiedon esitystekniikat. Tutkimuksen mukaan vastaanotetun tiedon soveltaminen toimii pohjana tulevaisuudessa hyödylliselle ajattelulle ja ongelmanratkaisulle, joissa korostuu ajatuksen tasolla tapahtuva syy–seuraus-suhteiden hahmottaminen (vrt. algoritmisen ajattelu, 2.1). (Salo ym., 2011, 26–28).

Viime vuonna tutkimukset osoittivat samansuuntaisia tuloksia. Ohjelmointi, robotiikka sekä automaatio nousevat esiin tulevaisuuden peruspilareina. Näihin liittyy vää ymmärrystä pidetään tärkeänä osana teknologista lukutaitoa, jota tarvitaan niin arkielämässä kuin työpaikoillakin. (Lindfors & Pirttimaa, 2018, 3.) Wing (2006, 35) esittääkin, että ohjelmoinnin avulla harjoiteltava ohjelmoinnillinen ajattelu (engl. *computational thinking*) on perustavanlaatuinen taito 2000-luvulla meille kaikille. Käsitteelle on esitetty muun muassa seuraavia suomennoksia: algoritmisen ajattelu, ohjelmoinnillinen ajattelu, automatisointiajattelu ja laskennallinen ajattelu. Tässä tutkimuksessa käytän näistä ensimmäistä – algoritmista ajattelua – sillä opetussuunnitelmassa käytetään samaa ja tutkimuskohteenani on ohjelmoinnin täydennyskoulutuksen käyneet luokanopettajat.

Edelliseen nojaten on ymmärrettävää, että myös opetussuunnitelman on pysyttävä muutoksessa mukana taataksemme oppilaiden pärjäämisen myös tulevaisuudessa. Sanotaan, että tietotekniikan kehitys on eksponentiaalista. Se tarkoittaa sitä, että kehitys tapahtuu kiihtyen, ottaen välillä huimiakin harppauksia kerralla. (Ford, 2015, 79–80.) Uusin Perusopetuksen opetussuunnitelman perusteet julkaistiin vuonna 2014 ja alakouluissa se astui voimaan syksyllä 2016. Tämä

uusi opetussuunnitelma korostaakin laaja-alaista osaamista, jonka tarve nousee nimenomaan ympäröivästä yhteiskunnasta ja sen muutoksista (POPS 2014, 17). Yksi opetussuunnitelman seitsemästä laaja-alaisesta osaamiskokonaisuudesta on *Tieto- ja viestintäteknologinen osaaminen*, joka alakoulussa sisältää mm. ikäkauden huomioon ottavaa ohjelmointia erilaisissa ohjelmointiympäristöissä (POPS 2014, 21, 103, 261). Jo ensimmäisillä vuosiluokilla matematiikan sisällyksessä ja tavoitteissa ohjelmointi alkaa laatimalla vaiheittaisia toimintaohjeita sekä kokeilemalla niiden toimivuutta. Ajatuksena on, että oppilaat saavat kokemuksia siitä, miten voi itse omilla ratkaisuillaan vaikuttaa teknologian toimintaan. Myöhemmin 3.–6. luokilla oppilaan tulisi jo osata ohjelmoida toimiva ohjelma graafisessa ohjelmointiympäristössä. (POPS 2014, 135, 165, 265.) Ohjelmointi ei kuitenkaan esiinny opetussuunnitelmassa omana oppiaineenaan, vaan sitä tulee sisällyttää muihin oppiaineisiin, ainakin matematiikan ja käsityön opetukseen. Yläkoulun puolella yksi ohjelmoinnin tavoitteista on algoritmisen ajattelun kehittymisen (POPS 2014, 430). Samalla kun ohjelmointi tuli osaksi opetussuunnitelmaa, tarve opettajien täydennyskoulutukselle kasvoi.

Yhä enenevässä määrin kouluilta odotetaan sellaisten valmiuksien tarjoamista, joiden avulla oppilaat pärjäävät digitaalisen yhteiskunnan kansalaisina. Tämän toteuttamiseksi tarvitsemme sellaisia opettajia, jotka ovat osaavia ja motivoituneita tukemaan oppilaita tulevaisuuden taitojen oppimisessa. (Hatlevik & Hatlevik, 2018.) Näin ollen opettajat tarvitsevat jatkuvaa täydennyskoulutautumista. Valtioneuvoston Digiajan peruskoulu -hankkeen aineistonkeruuvuosina 2017–2018 yksi opettajien heikoiden suorittamista osa-alueista oli ohjelmointi. (Tanhua-Piironen ym., 2019, 13–14, 21.) Aikaisemmissa tutkimuksissa on keskitytty tähän lähinnä opettajan osaamisen, opetuksen määrän ja koetun laadun näkökulmista (ks. Tanhua-Piironen ym., 2019; Hietikko ym., 2016). Vuorisen (2019, 112) mukaan nyt tarvitaankin parempaa ymmärrystä siitä, mitä opettajat itse ajattelevat tieto- ja viestintäteknologiasta (sis. ohjelmointi).

Tämän tutkimuksen kannalta oleellinen täydennyskoulutuksen tarjoaja ohjelmoinnin saralla on Innokas-verkosto, jonka toimintaa koordinoi Helsingin yliopiston Kasvatustieteellinen tiedekunta. Verkosto toimii yhteistyössä aluekoordinaat-

torikuntien lisäksi myös lukuisten muiden sidosryhmien kanssa. (Innokas Network, 2019.) Tämän tutkimuksen aineisto on kerätty Innokas-verkoston järjestämien täydennyskoulutuksien yhteydessä.

Tässä tutkimuksessa olen kiinnostunut siitä, mitä luokanopettajat itse ajattelevat siitä, miten he toteuttavat ohjelmoinnin opetusta ennen täydennyskoulutusta, sen jälkeen sekä tulevaisuudessa. Tutkimustehtävänäni on analysoida ja tulkita täydennyskoulutuksen vaikutuksia luokanopettajien ohjelmoinnin opetukseen.

Seuraavassa luvussa käsittelem tarkemmin peruskoulussa toteutettavaa ohjelmointia niin algoritmisen ajattelun, opetussuunnitelman ja ohjelmoinnin opettamisen kuin opetusmateriaalien ja -ympäristöjenkin kautta.



## 2 Ohjelmointi peruskoulussa

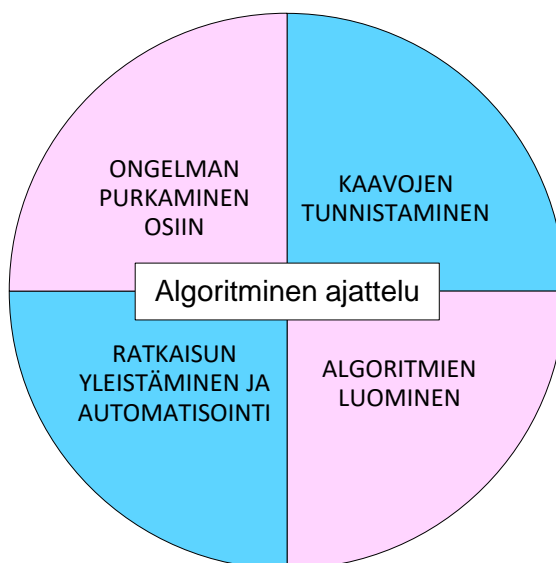
Ohjelmoinnin kaltaista toimintaa esiintyy arjessamme lähes päivittäin. Esimerkiksi jälkiruokareseptin vaiheiden selostaminen käy alkeellisesta ohjelmoinnista. (Hyvönen, Lappalainen & Lakanen, 2013, 2.) Yksinkertaisesti ohjelmoinnin voi ajatella olevan vain ohjeiden antamista tietokoneelle (Liukas & Mykkänen, 2014, 17). Tässä luvussa käsittelen ohjelmointia opetussuunnitelmassa sekä ohjelmoinnin opettamista. Ohjelmoinnin opettamiseen liittyy tiiviisti erilaiset ohjelmointiympäristöt, robotit sekä ohjelmointikielet, joista esittelen tutkimukseni kannalta oleellisimpia. Lopuksi käsittelen vielä kokoavasti luvun antia. Edellä mainittujen hahmottamisen helpottamiseksi aloitan avaamalla käsitteen *algoritminen ajattelu*.

### 2.1 Algoritminen ajattelu

Algoritminen ajattelu, ohjelmoinnillinen ajattelu, automatisointiajattelu, laskennallinen ajattelu – esimerkiksi tällaisia suomenoksia käytetään ohjelmoinnin kentällä englanninkielisestä termistä *computational thinking* (Kekäläinen 2015; POPS 2014; Wing, 2006). Koska tutkimukseni sijoittuu peruskoulukontekstiin ja opetussuunnitelmassa käytetään termiä algoritminen ajattelu, koen sen olevan relevantein termi myös tässä tutkimuksessa. Standlin (2017, 3) mukaan algoritmisessa ajattelussa määritellään, ymmärretään ja ratkaistaan ongelmia sekä perustellaan niitä. Lisäksi automaation ymmärrys ja soveltaminen ovat olennainen osa algoritmista ajattelua (Standl, 2017, 3). Seuraavaksi esittelen muun muassa Selbyn ja Woollardin (2014) sekä Mykkäsen ja Liukkaan (2014) käyttämän määritelmän algoritmisesta ajattelusta, jonka jälkeen avaan myös Futschekin (2006) käsitystä käsitteestä. Aiheeseen liittyvissä julkaisuissa on usein käytetty juuri näitä malleja, joten siksi esittelen myös tässä juuri ne.

Algoritmilla tarkoitetaan kuvausta tai ohjetta, jota seuraamalla voidaan ratkaista jokin tietty ongelma. (Peda.net, 2019; Liukas & Mykkänen, 2014, 21.) Algoritmissen ajattelun Selby ja Woollard (2014) kuvaavat koostuvan neljästä eri osa-alueesta, joita myös Liukas ja Mykkänen (2014, 76–78) ovat käyttäneet ohjelmointia opettaville opettajille suunnatussa oppaassaan (kuvio 1): *ongelman purkaminen*

*osiin; kaavojen tunnistaminen; algoritmien luominen sekä ratkaisun yleistäminen ja automatisointi.*



Kuvio 1: Algoritminen ajattelu (Selby & Woollard, 2014; Liukas & Mykkänen, 2014, 76–78).

*Ongelman purkaminen osiin* tarkoittaa esimerkiksi sitä, että hajotetaan luku 329 satoihin, kymmeniin ja ykkösiin:  $3 \times 100 + 2 \times 10 + 9 \times 1$ . Toisaalta minkä tahansa ongelman voi purkaa osiin, jotta sitä olisi helpompi hahmottaa, käsitellä ja ratkaista. Tällaista taitoa tarvitaan niin arjessa kuin ohjelmoinnissakin, kun jokin ongelma on hieman liian monimutkainen ratkaistavaksi kerralla.

*Kaavojen tunnistaminen* on yksinkertaisimmillaan helppojen syy–seuraus-suhteiden hahmottamista.

*Algoritmien luominen* tarkoittaa kuvauksen tai ohjeen luomista. Tällainen voi olla esimerkiksi reseptin kirjoittaminen kohta kohdalta.

*Ratkaisun yleistäminen ja automatisointi* on algoritmisen ajattelun neljäs osa-alue. Esimerkiksi matemaattisiin kaavoihin liittyy tällaista ajattelua: ihminen joutuu sijoittamaan luvut kaavaan yhä uudestaan, kun taas koneelle riittää opettaa kaava kerran, jonka jälkeen ratkaisut syntyvät kuin itsestään. (Selby & Woollard, 2014; Liukas ja Mykkänen, 2014, 76–78.)

Futschek (2006, 160) määrittelee algoritmisen ajattelun hieman Liukasta ja Mykkästä (2014) seikkaperäisemmin. Hänen mukaansa algoritminen ajattelu koostuu yhteensä kuudesta kyvystä (kuvio 2).

Kyky...	...analysoida ongelmaa.
	...määritellä ongelma täsmällisesti.
	...keksiä perusmekanismi tiettyyn ongelmaan.
	...rakentaa oikea algoritmi perusmekanismia käyttäen ongelman ratkaisemiseksi.
	...ymmärtää ongelmaa erilaisissa tilanteissa.
	...kehittää algoritmin tehokkuutta.

Kuvio 2: Algoritminen ajattelu Futschekin (2006, 160) mukaan.

Futschekin (2006, 160) mukaan algoritmiseen ajatteluun kuuluu ongelman täsmällinen määrittely sekä analysointi. Nämä kyvyt voisivat olla verrattavissa Liukkaan ja Mykkäsen (2014, 77; Selby & Woollard, 2014) mallin osa-alueeseen *ongelman purkamisen osiin*. Samoin voisi nähdä, että Liukkaan ja Mykkäsen (2014, 77; Selby & Woollard, 2014) *kaavojen tunnistaminen* voisi Futschekin (2006, 160) näkemyksessä vastata kykyä keksiä perusmekanismi tiettyyn ongelmaan. Edelleen Liukkaan ja Mykkäsen (2014, 77–78; Selby & Woollard, 2014) *algoritmien luominen* muistuttaa Futschekin (2006, 160) mallin kykyä hyödyntää perusmekanismia oikean algoritmin rakentamiseksi. Ongelman ymmärtämisen erilaisissa tilanteissa (Futschek, 2006, 160) taas voisi rinnastaa Liukkaan ja Mykkäsen (2014, 78; Selby & Woollard, 2014) *ratkaisun yleistämiseen*. Kuvattujen määritelmien perustavanlaatuisin eroavaisuus lienee Mykkäsen ja Liukkaan (2014; Selby & Woollard, 2014) versiosta puuttuva mutta Futschekin (2006, 160) mallista löytyvä algoritmin tehokkuuden kehittäminen.

Molempia edellä kuvattuja määritelmiä vahvistaa Wingin (1006, 33) näkemys siitä, että algoritmiseen ajatteluun tarvitaan niin ongelmanratkaisukykyä kuin järjestelmällisyyttäkin. Myös Tedren ja Denningin (2016, 120) mukaan algoritminen ajattelu on kokoelma algoritmisia ideoita sekä ajattelun taitoja. Seuraavassa luvussa perehdyntäisiin siihen, millä tavalla algoritminen ajattelu ja ohjelmointi esiintyvät opetussuunnitelmassa.

## 2.2 Ohjelmointi opetussuunnitelmassa

Kuten edellä totesin, algoritminen ajattelu sekä ohjelmointi esiintyvät nykyisessä perusopetuksen opetussuunnitelmassa (POPS 2014). Jotta niiden päätymistä osaksi opetussuunnitelmaa olisi helpompi hahmottaa, käyn lyhyesti läpi aiheen historiaa.

Elektroninen tietojenkäsittely (ETK) on termi ajalta, jolloin maailmaa rakennettiin ihmisten opittua hyödyntämään sähköä paitsi valon ja lämmön myös informaation kuljettajana. Esimerkiksi paperikoneiden toiminta mahdollistui juuri elektronisen tietojenkäsittelyn tuloksena. Elektronisen tietojenkäsittelyn aikakautta seurasi automaattisen tietojenkäsittelyn (ATK) aika. (Rousku ym., 2019, 26.) Saarikosken (2006, 5) mukaan vuosina 1987–1995 luotiin raamit ja pohja suomalaiselle ATK-opetukselle. Käytännössä tietokoneopetus yleistyi ensin kokeilukoulujen ja ATK-kerhojen voimasta. Esimerkiksi Tapiolan yhteiskoulussa oli mahdollista valita kerhomuotoista ATK-opetusta jo vuonna 1965 ja 1970-luvulla tällaisia kokeiluja suoritettiin vakituisemmin jo muuallakin. Vuonna 1977 lähdettiin selvittämään tietokoneiden opetuskäyttöön liittyviä haasteita ja mahdollisuuksia tausta-ajatuksena vakinaistaa tietotekniikan opetus niin peruskoulussa kuin lukiossakin. Tausta-ajatus toteutui 1980-luvun lopulla, kun tietokoneopetus sisällytettiin osaksi koulujen opetussuunnitelmia. (Saarikoski, 2004, 92–93; Saarikoski, 2006, 5–7.) Nyt maailmamme on siirtymässä tekoälyaikaan, jolloin myös tietojenkäsittely muuttuu älykkääksi (ÄTK) (Rousku ym., 2019, 26). Kaikki tämä yhdistettynä tiedon hallinnan merkityksen kasvuun aiheuttaa tarpeen suhtautua informaatioon uudella tavalla. Tällöin olennaisina asioina voidaan pitää tiedon etsimistä, käsittelyä ja tuottamista niin sosiaalisessa, teknologisessa kuin visuaalisessakin ympäristössä. (Halinen, 2009, 36). Ohjelmointi ja algoritminen ajattelu esiintyvätkin tämän tyylinä myös opetussuunnitelmassa.

Ohjelmointi esiintyy opetussuunnitelmassa (POPS 2014) osana tieto- ja viestintäteknologian (tvt) osaamista. Tieto- ja viestintäteknologia on yksi opetussuunnitelman seitsemästä laaja-alaisen osaamisen tavoitteista. Tämän lisäksi ohjelmointi on sisällytetty osaksi matematiikan ja käsityön tavoitteita ja sisältöjä (taulukko 1).

Taulukko 1: Ohjelmointi opetussuunnitelmassa (POPS 2014).

	Luokat 1–2	Luokat 3–6	Luokat 7–9
<i>Laaja-alainen osaaminen: Tieto- ja viestintätekologinen osaaminen</i>	Ikäkaudelle sopiva <b>ohjelmointi</b>	<b>Ohjelmoinnin</b> avulla ymmärretään, miten teknologian toiminta riippuu ihmisen tekemistä ratkaisuista	<b>Ohjelmointi</b> osana eri oppiaineita
<i>Matematiikka</i>	Ajattelun taidot: Tutustuminen <b>ohjelmoinnin</b> alkeisiin laimalla ja testaamalla vaihteittaisia toiminta-ohjeita	Ajattelun taidot: Graafisessa <b>ohjelmointi</b> ympäristössä ohjelmien suunnittelu ja toteuttaminen  6. luokan hyvän arvosanan kriteerit: "Oppilas osaa ohjelmoida toimivan ohjelman graafisessa <b>ohjelmointi</b> ympäristössä"	Ajattelun taidot: <b>Algoritmis</b> en ajattelun syventäminen ja <b>ohjelmoinnin</b> käyttäminen ongelmien ratkaisemiseksi  9. luokan hyvän arvosanan kriteerit: "Oppilas osaa soveltaa <b>algoritmis</b> en ajattelun periaatteita ja osaa <b>ohjelmoida</b> yksinkertaisia ohjelmia"
<i>Käsityö</i>	-	<b>Ohjelmoidaan</b> toimintoja, esimerkiksi robotiikka ja automaatio	<b>Ohjelmoinnin</b> soveltaminen suunnitelmiin sekä valmistettaviin tuotteisiin

Tarkastelen seuraavaksi lähemmin taulukkoa 1. Matematiikassa ohjelmoinnin tulee olla osana opetusta myös läpi peruskoulun. Se sisällytetään jokaisessa vuosiluokkaryhmässä ajattelun taitoihin. (POPS 2014.) Liukas ja Mykkänen (2014, 57) näkevät asian samalla tavalla. Ohjelmointi voidaan nähdä keinona oppia kognitiivisia taitoja, kuten luovaa ajattelua, tarkkaa työskentelyä, kykyä hahmottaa ongelma ja valmiuksia keksiä sille erilaisia ratkaisuja (Liukas & Mykkänen, 2014, 57). Vuosiluokilla 7–9 matematiikan opetussuunnitelmaan on nostettu mukaan algoritminen ajattelu (POPS 2014, 435). Taulukosta 1 voi nähdä ajattelun taitojen kehittymisen jonkinlaisena jatkumona: aloitetaan alkeista ja yksinkertaisista algoritmeista, sitten harpataan suunnittelemaan ja toteuttamaan ohjelmia erilaisissa

graafisissa ohjelmointiympäristöissä ja lopulta syvennetään osaamista entisestään. Lisäksi ohjelmointi ja algoritmien ajattelu ovat osa sekä 6. että 9. luokan arviointien hyvän arvosanan kriteerejä.

Ohjelmointi on myös osa käsityön tavoitteita peruskoulussa, mutta vasta kolmannelta luokasta eteenpäin. Alakoulussa ohjelmoidaan erilaisia toimintoja esimerkiksi robotiikan avulla. Yläkoulussa tavoitteisiin kuuluu ohjelmoinnin soveltaminen suunnitelmiin sekä valmistettaviin tuotteisiin. (POPS 2014.) Ohjelmointi voidaan helposti myös yhdistää mihin tahansa oppiaineeseen laaja-alaisten osaamistaitojen kautta, sillä opetussuunnitelmassa (POPS 2014) laaja-alaisten osaamistavoitteiden sisältä löytyvät maininnat peruskoulun opetussuunnitelmassa jokaisen vuosiluokkaryhmän (1–2, 3–6 ja 7–9) kohdalla. Isommassa kuvassa laaja-alaisessa osaamisessa on kyse siitä, että ihminen selviytyy erilaisista tilanteista kulloinkin tarvittavia asenteita, tietoja ja taitoja luovasti hyödyntäen. Tärkeänä laaja-alaista osaamista pidetään siksi, että yhteiskunnassa tarvittavat tiedot ja taidot muuttuvat jatkuvasti, eikä muutokseen liene parempaa tapaa valmistautua kuin harjoitella sellaista osaamista, mitä voidaan soveltaa tilanteessa kuin tilanteessa. (Halinen, 2009, 36, 38.)

Seuraavaksi esittelen ohjelmoinnin opettamista ja siihen liittyvää tutkimusta, joka toimii siltana täydennyskoulutuksen käsittelemiseen.

### **2.3 Ohjelmoinnin opetus**

Ohjelmoinnin opettamisen onnistumiseksi esitetään erilaisia edellytyksiä. Ensinnäkin laitteiden olemassaolo ja toimivuus ovat edellytyksiä ohjelmoinnin opetuksen onnistumiselle (Vuorinen, 2019, 106) ainakin 3. luokasta ylöspäin, sillä sitä alemmilla luokilla ohjelmoinnin opetus voi tapahtua myös algoritmien harjoitteluna ilman laitteita (POPS 2014). Valtioneuvoston selvityksen (Tanhua-Piironen 2019, 18) mukaan opettajat ovat keskimäärin jokseenkin tyytyväisiä toimintaympäristöihinsä laitteiden saatavuuden ja toimivuuden saralla.

Toinen edellytys liittyy pedagogiikkaan: Vuorisen (2019, 104–105) näkemyksen mukaan tieto- ja viestintäteknologian hyödyntäminen opetuksessa riippuu pitkälti

siitä, tukeeko se opettajien mielestä pedagogista näkökulmaa sekä yhteiskunnasta kumpuavia odotuksia ja arvoja. Myös Norrena (2015) tuo esille pedagogisen näkökulman tärkeyden teknologian hyödyntämisessä. Hän esittääkin tieto- ja viestintäteknologian tuovan opetukseen pedagogista lisäarvoa silloin, kun oppimistehtäviä tuotetaan oppilaan itsensä tuottamina (Norrena, 2015, 70).

Tainan (2015, 2) mukaan oppilaat hyötyvät opetuksesta eniten silloin, kun heidän annetaan käyttää mielikuvitusta, mahdollistetaan konkreettisten asioiden tekeminen sekä omien töidensä tuloksien näkeminen. Myös Norrenalla (2015) on samankaltaisia ajatuksia oppilaiden osallistamisesta. Lisäksi hän esittää, että teknologian hyödyntäminen kohdistuisi tiedon rakentamiseen ja ongelmanratkaisuun (Norrena, 2015, 235). Ja näitä asioita ohjelmointi parhaimmillaan onkin.

Aiemmin esittelin algoritmisen ajattelun osa-alueet (Selby & Woollard, 2014; Liukas & Mykkänen, 2014, 76–78) (ks. 2.1). Liukkaan ja Mykkäsen (2014, 78) mukaan ohjelmoinnin opettaminen on pohjimmiltaan juuri näitä algoritmiseen ajatteluun sisältyvien taitojen opettamista: ongelman purkamista osiin, kaavojen tunnistamista, algoritmien luomista sekä ratkaisun yleistämistä ja automatisointia. Futschek (2006, 160) näkee algoritmisen ajattelun opettamisen määrittelemisen hieman haastavampana. Hänen mukaansa yksinkertaistettu määritelmä algoritmisen ajattelun opettamiselle voisi olla monien eri ongelmien ratkaisemisen opettaminen. Ongelma tulisi kuitenkin valita huolellisesti, sillä liian yksinkertaiset ongelmat eivät välttämättä anna tilaa luovuudelle, yksilöllisille oppijoille sekä epätavallisille ratkaisuille. (Futschek, 2006, 160.)

Seuraavaksi esittelen vielä tutkimukseni kannalta oleellisimpia ohjelmointiympäristöjä, robotteja ja kieliä.

### **Ohjelmointiympäristöt, robotit ja ohjelmointikielet**

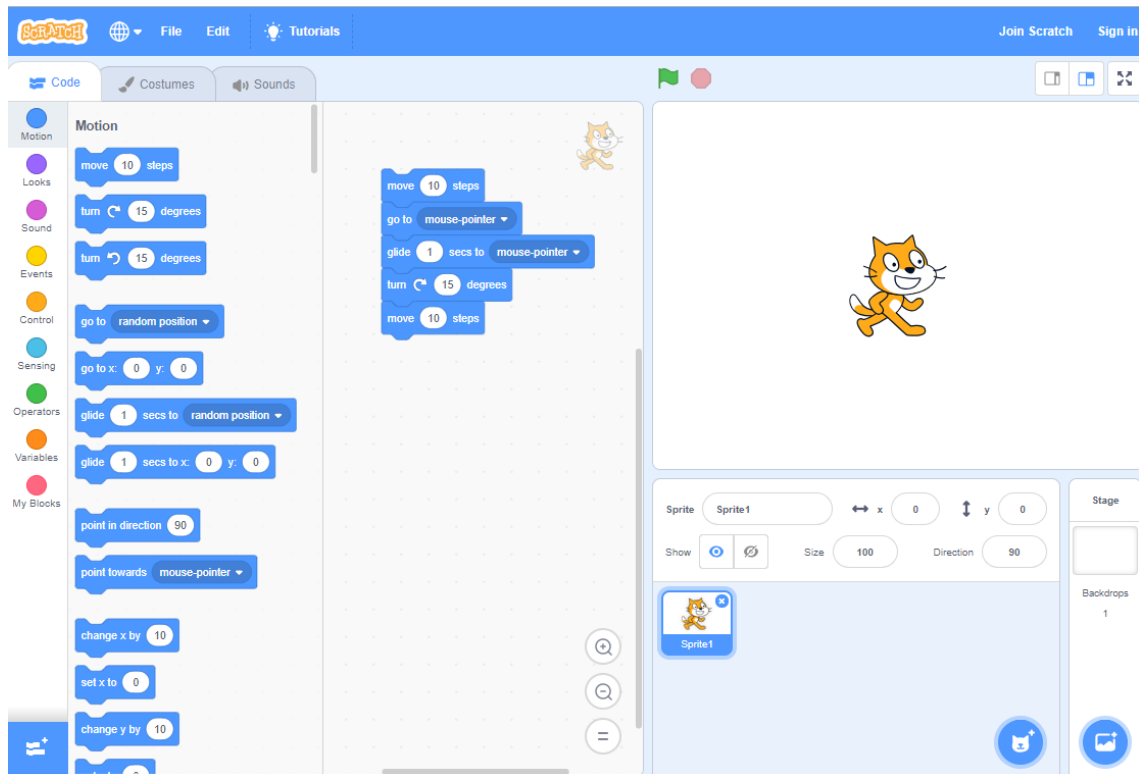
Ohjelmointiympäristöt, robotit ja ohjelmointikielet eivät ole uusia asioita oppilaitoksissa, vaan niitä ja niiden käyttöä on myös tutkittu yli 30 vuoden ajan (Blikstein,

2015, 2). Blikstein (2015) esittääkin, että koulutuksessa käytetty lasten ohjelmointikielien kehittäminen on jaettavissa viiteen eri sukupolveen alkaen 1980-luvun Lego-roboteista 2010-luvun älylaitteisiin.

Graafiset ohjelmistoympäristöt mainitaan opetussuunnitelmassa 3.–6. luokkien matematiikan opetuksen kohdalla (POPS 2014, 261). Valtioneuvoston selvityksen mukaan (Tanhua-Piironen, 2019, 28) mukaan vain noin 22 prosenttia vastaajista on kokeillut oppilaiden kanssa graafista ohjelmointia. Opettajat itse arvioivat graafisen ohjelmoinnin osaamisensa olevan heikkoa, mutta tulosten mukaan opettajien osaaminen on kuitenkin kasvanut edelliseen vuoteen 2018 verrattuna (Tanhua-Piironen, 2019, 28).

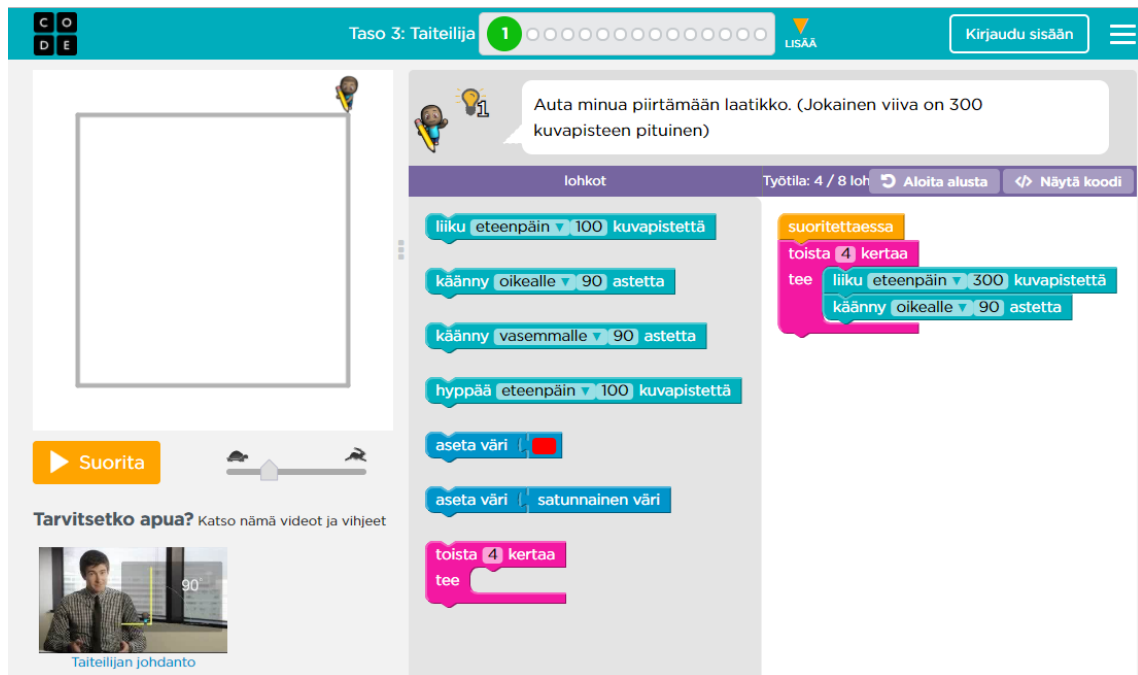
Tutkimukseni kannalta oleelliset graafiset ohjelmointiympäristöt ovat Scratch, Scratch Junior sekä Code.org, sillä ne nousevat aineistosta esille voimakkaimmin. Scratch (kuva 1), sekä sen nuoremmille suunnattu versio Scratch Junior, ovat ilmaisia ja avoimia kaikille. Ne on kehitetty ja suunnattu lapsille siten, että ne mahdollistaisivat innostavan tavan tutustua algoritmiseen ajatteluun. Tekijöiden mukaan Scratchin avulla oppii ohjelmoinnin perusteet nopeasti. Sen avulla pystyy luomaan muun muassa vuorovaikutteisia tarinoita, sarjakuvia, pelejä sekä musiikillisia elementtejä. (Nelimarkka, Vainio, Kinnunen, 2011, 3; Quahbi, 2015, 1480.)





Kuva 1: Scratch. Kuvakaappaus (<https://scratch.mit.edu/projects/editor/?tutorial=getStarted>).

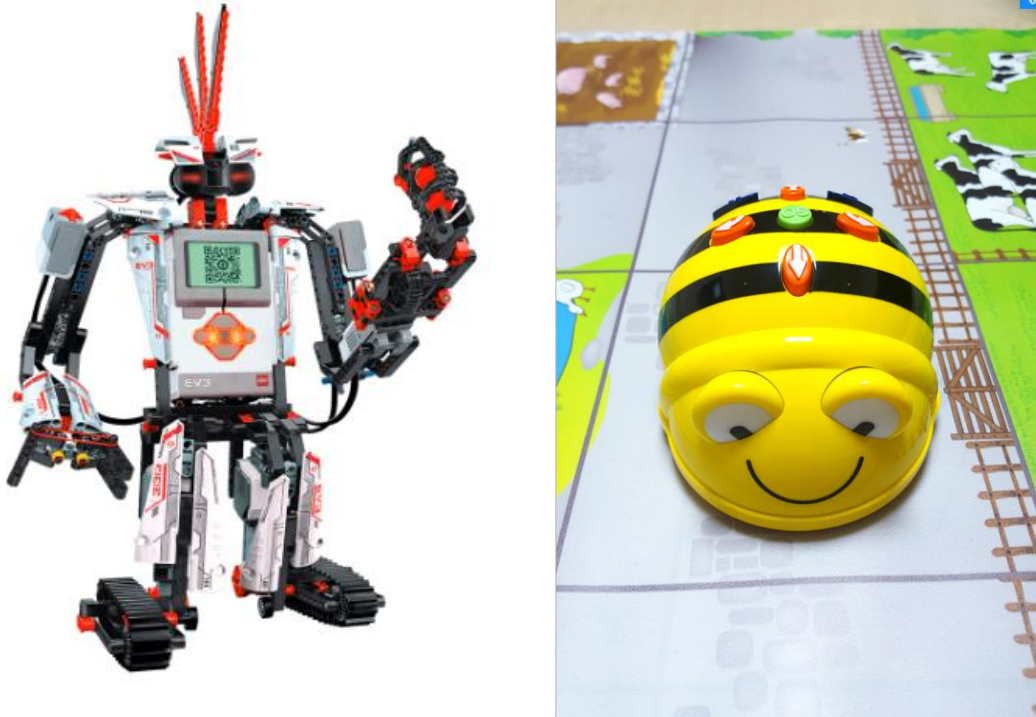
Code.org (kuva 2) on puolestaan selainpohjainen ohjelmointiympäristö, joka sisältää esimerkiksi erilaisia graafisia pelejä. Code.org itse kirjoittaa verkkosivulleen olevansa voittoa tavoittelematon yritys tavoitteenaan laajentaa ohjelmoinnin saatavuutta kouluissa. (Code.org, 2019.) Tässä tutkimuksessa nousevat myös esille Koodaustunti- sekä Hour of Code -sivustot. Nämä eivät varsinaisesti ole sama asia kuin Code.org, mutta molemmilla sivuilla tehtäviä klikatessa selain ohjaa ohjelmoijan Code.org-sivustolle. Tämän vuoksi käytän tästä eteenpäin näistä kaikista käsitettä Code.org.



Kuva 2: Code.org. Kuvakaappaus (<https://studio.code.org/s/course3/stage/3/puzzle/1>)

Tutkimuksessani oleellisimpia robotteja ovat Bee-Bot, Blue-Bot sekä erilaiset Lego-robotit (kuva 3). Bee-Bot ja sen kehittyneempi versio Blue-Bot ovat lattiarobotteja, joiden avulla tutustutaan ohjelmoinnin perusteisiin. Nämä robotit voidaan ohjelmoida kulkemaan kaikkiin pääilmansuuntiin esimerkiksi lattiaan teipatussa ruudukossa tai erikseen ostettavilla matoilla. Näiden robottien hinnat vaihtelevat noin 80 eurosta 120 euroon. (Erilaiset oppijat, 2019; Lekolar, 2019.) Myös Lego valmistaa erilaisia robotteja. Esimerkiksi Lego Mindstorms EV3 -robotti kuuluu ensin rakentaa, jonka jälkeen sitä voi ohjata ohjelmointikäyttöliittymällä, joka perustuu helppokäyttöisiin kuvakkeisiin. Tämän tyyppisten Lego-robottien hinnat vaihtelevat 120 eurosta 300 euroon. (LEGO.com, 2019; verkkokauppa.com, 2019.)

Tässä tutkimuksessa ainoat esille nousevat robotiikkaan liittyvät ohjelmointikielet ovat ROBOLAB- ja NXT-G-kielet. ROBOLAB on kuvallinen ohjelmointikieli, joka oli käytössä Lego-roboteissa ennen kuin NXT-ohjelmisto julkaistiin. NXT onkin ikään kuin ROBOLAB-kielen kehittynyt versio. (LEGO Engineering, 2019.)



Kuva 3: Ev3rstorm-legorobotti ja Bee-Bot-robotti. Kuvakaappaukset (<https://www.lego.com/fi-fi/mindstorms/build-a-robot/ev3rstorm> & <https://www.publicdomainpictures.net/en/view-image.php?image=143737&picture=bee-bot>)

## 2.4 Yhteenveto

Tässä luvussa lähdin liikkeelle algoritmista ajattelusta ja päädyin opetussuunnitelman kautta ohjelmoinnin opettamiseen. Algoritminen ajattelu sekä ohjelmoinnin opettaminen ovat vahvasti kytköksissä toisiinsa, sillä ne ovat osittain myös hieman päällekkäiset – algoritmista ajattelua hyödynnetään ohjelmoitaessa ja ohjelmoitaessa algoritminen ajattelu kehittyy. Ilman toista ei siis ole toista. Algoritminen ajattelu ja ohjelmoinnin opettaminen ovat osa opetussuunnitelmaa, joka antaa suuntaviivat ohjelmoinnin opettamiselle peruskoulussa.

Mutta miten opettajat pysyvät muutoksien perässä ja oppivat opettamaan ohjelmointia? Tähän kysymykseen etsin vastauksia seuraavassa luvussa, jonka aiheena on täydennyskoulutuksen vaikutukset.

### 3 Täydennyskoulutus

Kuten muussakin opetuksessa, myös ohjelmoinnin opetuksen onnistumisessa opettajalla on suuri rooli. Koska teknologinen miljö ja sen hallitsemisen taitovaatimukset muuttuvat nopeasti (Ford, 2015, 79), jatkuvan täydennyskoulutuksen tarve on ilmeinen (Tanhua-Piironen ym., 2019, 13–14). Mikkolan (2012, ix) mukaan opettajan pätevyys edellyttää perus- ja täydennyskoulutuksen jatkuvuutta. Tässä luvussa esittelen ensin täydennyskoulutuksen tarvetta ja sen muotoja, jonka jälkeen siirryn täydennyskoulutuksen vaikuttavuuteen.

#### 3.1 Tarve ja muodot

Valtioneuvoston tänä vuonna tekemän selvityksen (Tanhua-Piironen ym., 2019, 21) mukaan tieto- ja viestintätekniikan taidoista opettajat suoriutuivat heikoiten tietoverkkojen, asennettavien sovellusten turvallisuuden arvioinnin, kuvankäsittelyn ja tietokoneiden perustoiminnallisuuksien hallinnan lisäksi ohjelmoinnissa. Selvityksen mukaan opettajat siis kaipaavat lisäkoulutusta ohjelmointiin liittyen (Tanhua-Piironen, 2019, 38). Opettajien heikot suoritukset ja merkittävät tasoerot ohjelmointitaidoissa herättävät kysymyksen siitä, voidaanko kaikille oppilaille taata tasa-arvoiset lähtökohdat ohjelmoinnin ja algoritmisen ajattelun kartuttamiseksi (Muhonen, Kaarakainen & Savela, 2015).

Opetusalan Ammattijärjestön (OAJ) vuonna 2016 kirjoitetun julkaisun (Hietikko, Ilves & Salo, 2016, 16) mukaan tieto- ja viestintätekniikan täydennyskoulutuksissa nähtiin puutteita niin niiden määrän kuin riittävyydenkin puolesta. Tuolloin OAJ:n teettämän kyselyn mukaan täydennyskoulutusta oli kyllä saatu, mutta opettajien itsensä mielestä koulutus ei ollut ollut riittävää (Hietikko ym., 2016, 17). Tänä vuonna Valtioneuvosto (Tanhua-Piironen ym., 2019) ehdotti erilaisia toimenpiteitä tilanteen kohentamiseksi. Toimenpide-ehdotukset pitävät sisällään kehotuksen kehittää koulujen digitutorointia siten, että se tukisi opettajien teknologista ja pedagogista osaamista. Lisäksi ehdotuksissa nostetaan esille digitaalisten mahdollisuuksien oppilaslähtöisen opetuksen tärkeys. (Tanhua-Piironen ym., 2019, 1.) Tällainen oppilaslähtöinen opetus pitää sisällään niin oman sisällön tuottamisen kuin jakamisenkin, mikä itsessään voidaan rinnastaa esimerkiksi ohjelmoinnin opettamiseen.

Täydennyskoulutuksessa tarpeelliseksi nähtyjä sisältöjä ovat esimerkiksi algoritmisen ajattelun tukeminen sekä teknologian yleinen ymmärtäminen ja hyödyntäminen. Algoritmiseen ajatteluun keskittyttäessä voidaan paneutua esimerkiksi painottamalla tieto- ja viestintäteknologian opettamista ylipäänsä tai hyödyntämällä teknologiaa pedagogisessa mielessä ainerajoja ylittävästi. (Vuorinen, 2019, 92.) Vuorisen (2019, 93) mukaan täydennyskoulutuksia suunniteltaessa on myös oleellista pohtia sitä, mikä on täydennyskoulutuksen kohderyhmä. Valtioneuvoston selvityksen mukaan kaikkein tärkein kohderyhmä olisi tällä hetkellä yli 40-vuotiaat opettajat sekä erityisopettajat. Tutkimukseni kannalta mielenkiintoista myös on, että selvityksen mukaan kaikista tutkituista opettajista (aineenopettajat, erityisopettajat, luokanopettajat, maahanmuuttajaopettajat ja opinto-ohjaajat) juuri luokanopettajat arvioivat itse ohjelmointiin liittyvän osaamisensa olevan parempaa verrattuna muihin opettajaryhmiin. (Tanhua-Piironen ym., 2019, 1, 28.)

Täydennyskoulutukseen liittyy myös haasteita. Esimerkiksi Vuorinen (2019, 105) nostaa esille täydennyskoulutuksien päämäärien saavuttamisen ongelmallisuuden silloin, kun opettajien teknologinen osaaminen on niin heikkoa, että se rajoittaa itse ohjelmoinnin oppimista. Hän ehdottaakin, että ohjelmoimisen tulisi olla vain sellaisten opettajien tehtävänä, joilla on riittävät edellytyksen opetella teknologiaa (Vuorinen, 2019, 105). Kysymysmerkiksi Vuorisen avauksessa jää se, keitä ja millaisin ominaisuuksin varustettuja nämä riittävän edellytyksen omaavat opettajat oikein ovat.

### **3.2 Vaikuttavuus**

Korkeakoski (1999,26) näkee, että täydennyskoulutuksen vaikuttavuudessa keskeisintä on sen opetuksen kehittämisen tukeminen ja sitä kautta opetuksen laadun paraneminen. Valtioneuvoston selvityksen (Tanhua-Piironen ym., 2019, 27) mukaan täydennyskoulutuksen käyneiden opettajien alkeisohjelmoinnin taidot kohentuivat vuosien 2017 ja 2018 välillä. Kohentuminen oli erityistä naisten ja luokanopettajien taidoissa. Tätä pidetään hyvänä tuloksena, sillä kyseiset ryhmät sijoittuivat alkeisohjelmoinnin heikoimmin hallitseviin. Tulokset ovat yhteydessä

arjessa käytettyjen digi- ja ohjelmistopalveluiden käyttöaktiivisuuteen sekä koettuun luottamukseen omia taitoja kohtaan. Lisäksi myös digiaiheisella täydennyskoulutuksella näyttää olevan positiivista vaikutusta. Ohjelmointiaiheisen täydennyskoulutuksen osalta vaikutukset näyttävät myös varsin positiivisilta. (Tanhua-Piironen ym., 2019, 27, 29.) OAJ:n julkaisun (Hietikko ym., 2016, 20) mukaan täydennyskoulutetut opettajat myös käyttävät tieto- ja viestintäteknologiaa muita opettajia aktiivisemmin ja monipuolisemmin. Seuraavaksi esittelen täydennyskoulutuksen hyötyjä hieman yksityiskohtaisemmin.

Tanhua-Piironen ja kumppaneiden (2019, 27) mukaan digitutorina toimivat opettajat hallitsevat alkeisohjelmoinnin osa-alueet huomattavasti kollegoitaan paremmin ja menestyivät myös tutkimuksen (Tanhua-Piironen ym., 2019, 23) yhteydessä tehtyjen tehtävien parissa menestyksekkäämmin. Fisherin, Byrnen ja Tangneyn (2016) Irlannissa toteuttamassa tutkimuksessa testattiin Bridge21-mallia. Kyseessä oli sosiaaliskonstruktivistinen malli, jonka keskiössä on tiimityöskentely sisältöä tuottaen. Tutkimuksen mukaan malli osoittautui tehokkaaksi, sillä tiimityöskentely mahdollisti vapaan keskustelun ideoista, kysymyksien esittämisen sekä ryhmän asiantuntijuuden hyödyntämisen. (Fisher ym., 2016.) Samantyyppisiä tuloksia on saatu myös Suomessa, kun täydennyskoulutus nähdään yksilötason osaamisen yhdistymisenä yhteisötason osaamiseen (Paaso, 2012, 50).

Siddigin ja Schererin (2016) mukaan opettajien luottamus omiin digitaitoihin on merkittävä edistäjä teknologian opetuskäyttöä ajatellen. Lisäksi on saatu selville, että etenkin iäkkäämpien opettajien joukossa itseluottamuksen merkitys kasvaa, sillä itseluottamuksen määrä on suoraan suhteessa opettajan halukkuuteen hyödyntää teknologiaan osana omaa opetustaan. (Siddig & Scherer, 2016.) Hietikon ja kumppaneiden tutkimuksen mukaan täydennyskoulutuksen käyneet opettajat näkivät digitalisaation innostavana ja he suhtautuivat myös omiin kykyihinsä itsevarmemmin (Hietikko ym., 2016, 20). Täydennyskoulutuksien hyötyjen kannalta tämä on hyvä uutinen, sillä kouluilta odotetaan yhä enenevässä määrin tarjoavan oppilaille sellaisia valmiuksia, joita he tarvitsevat tulevaisuudessa. (Hatlevik & Hatlevik, 2018.) Hatlevikin ja Hatlevikin (2018) mukaan tähän tehtävään tarvitaan

sellaisia opettajia, jotka ovat hyvän osaamistasonsa lisäksi myös motivoituneita oppilaiden tukemisessa.

Eräässä tutkimuksessa (Korkeakoski, 1999) kysyttiin täydennyskoulutuksen vaikuttavuudesta sadoilta opettajilta, jotka olivat käyneet erilaisia täydennyskoulutuksia. Tutkimuksen mukaan yli 90 % opettajista pitivät täydennyskoulutuksen vaikutuksia myönteisinä. Esille nousi myös, että koulutus oli onnistunut vahvistamaan opettajien pedagogisia taitoja. Merkittävää oli myös täydennyskoulutuksen vaikutuksen kohdistuminen opettajan omaan opetukseen, yhteistyöhön muiden opettajien kanssa tai jonkin oman koulun sisäisen hankkeen edistämiseen. Huomionarvoista myös oli, että tietotekniikan täydennyskoulutukseen osallistuneet opettajat olivat keskimääräistä tyytymättömpiä saamaansa koulutukseen ja erityisesti saamaansa pedagogiseen hyötyyn. (Korkeakoski, 1999, 70–71, 73, 113.)

Seuraavassa luvussa esittelen tutkimukseni tehtävän sekä tutkimuskysymykset.

## 4 Tutkimustehtävä ja tutkimuskysymykset

Tässä tutkimuksessa olen kiinnostunut siitä, millä tavalla luokanopettajat itse ajattelevat toteuttavansa ohjelmoinnin opetusta sekä millä tavalla täydennyskoulutus vaikuttaa luokanopettajien ohjelmoinnin opetukseen. Tutkimustehtävänäni onkin analysoida ja tulkita täydennyskoulutuksen vaikutuksia luokanopettajien ohjelmoinnin opetukseen. Lisäksi tarkoituksenani on selvittää, millä tavalla luokanopettajat itse ajattelevat toteuttavansa ohjelmoinnin opetusta ennen täydennyskoulutusta, sen jälkeen sekä tulevaisuudessa. Tämän pohjalta relevanteiksi tutkimuskysymyksiksi valikoituivat:

1. Millä tavalla luokanopettajat ajattelevat opettavansa ohjelmointia
  - a. ennen täydennyskoulutusta?
  - b. täydennyskoulutuksen jälkeen?
  - c. tulevaisuudessa?
2. Miten täydennyskoulutus vaikuttaa luokanopettajien ajatuksiin ohjelmoinnin opetuksesta?

Seuraavassa luvussa avaan tutkimukseni toteutusta.



## 5 Tutkimuksen toteutus

Tämän tutkimuksen tutkimuskohteena oli ohjelmointiin liittyvän täydennyskoulutuksen käyneet luokanopettajat ja heidän kokemuksensa täydennyskoulutuksen vaikutuksista heidän ohjelmoinnin opetukseensa. Tässä luvussa esittelen tutkimuksen toteutusta aloittaen tutkimusasetelmasta, edeten aineistoni kuvailuun ja aineiston analyysiin. Tutkimusasetelman ja aineiston esittelyä varten haastattelin Innokas-verkoston johtajaa Tiina Korhosta.

### 5.1 Tutkimusasetelma

Tutkimuksen aineisto kerättiin Innokas-verkoston järjestämien täydennyskoulutuksien yhteydessä. Innokas-verkoston toimintaa koordinoi Helsingin yliopiston Kasvatustieteellinen tiedekunta. Verkosto toimii yhteistyössä aluekoordinaattori-kuntien lisäksi myös lukuisten muiden sidosryhmien kanssa. (Innokas Network, 2019.) Tutkimuskohteenani olleet opettajat kävivät joko Kiinnostaako koodaus -kiertueen koulutuksen ja/tai Koodiaapiseen MOOC -koulutuksen.

Syksyllä 2015 Kiinnostaako koodaus -kiertue käsitti kaikkiaan koulutuksen järjestämisen yhdessätoista eri kunnassa Helsingistä Rovaniemelle. Koulutus oli yhden päivän mittainen, ja se sisälsi yhteisten luento-osuuksien lisäksi toiminnallisia työpajoja. Yhteisten luento-osuuksien aikana käytiin läpi, mitä ohjelmointi on, mitä innovaatiokasvatukseen kuuluu sekä millaisia ovat pedagogisesti mielekkäät ohjelmoinnin opetuksen ja oppimisen tavat. Lisäksi käytiin läpi ohjelmointia opetussuunnitelmassa: laaja-alaisen osaamisen taidot, algoritmisen ajattelun perusteet sekä monialaisten ja ilmiöpohjaisten oppimiskokonaisuuksien mahdollisuudet. Opettajat pääsivät myös kuulemaan digitutor-oppilastoiminnasta sekä tiimityöstä (esim. pedagoginen kahvila ja muu osaamisen jakaminen työyhteisössä). Päivän aikana opettajat osallistuivat kahteen neljästä työpajasta: Lähtölaukaus koodaukseen (ohjelmoinnin opettamisen perusteet), Sukellus pelien tekemiseen (Scratch), Rohkeasti robotiikkaan (Lego-robotit) sekä Racketilla kohti uusia ulottuvuuksia (Racket). Tämän koulutuksen rahoittajina toimivat Opetushallitus sekä Teknologiateollisuuden 100-vuotissäätiö. Koulutuspäivän päätteeksi opettajat täyttivät koulutusta koskevan kyselyn, josta tuli osa myös tämän tutkimuksen aineistoa.

Koodiaapisen MOOC -koulutus on verkkokurssi ohjelmoinnista ja automatisointiajattelusta, ja se on suunnattu peruskoulun opettajille sekä kaikille aiheesta kiinnostuneille. Kurssin aluksi opettaja valitsi opettamansa ikäryhmän ja oppiaineen mukaan itselleen sopivimman linjan, joka koostui 5–8 opiskelujaksosta, jotka kukin kestivät yhden viikon. Jaksot sisälsivät ohjelmoinnin käsitteitä, käytännön harjoituksia sekä pohdintatehtäviä, ja näihin oli varattava aikaa viikoittain puolesta tunnista kahteen tuntiin. Tämän koulutuksen rahoittajana toimi Opetushallitus.

## 5.2 Aineisto

Tutkimuksen aineisto koostui kahdesta luokanopettajilla teetetystä kyselylomakkeesta koskien Innokas-verkoston täydennyskoulutuksien vaikutuksia. Kyselylomakkeet kerättiin Innokas-verkoston toimesta, enkä itse osallistunut aineiston keräämiseen. Ensimmäiseen kyselylomakkeeseen opettajat vastasivat syksyllä 2015 heti täydennyskoulutuksen käytyään. Toiseen kyselylomakkeeseen opettajat vastasivat kesällä 2016, noin puoli vuotta ensimmäisen kyselyn ja täydennyskoulutuksen jälkeen. Jälkimmäinen kysely toimitettiin opettajille sähköpostin välityksellä.

Tutkimuskohteenani oli 101 luokanopettajaa, jotka olivat käyneet vähintään yhden Innokas-verkoston täydennyskoulutuksen. Suurin osa opettajista ( $n = 65$ ) oli käynyt Kiinnostaako koodaus -koulutuksen ja muutama opettajista ( $n = 4$ ) oli osallistunut Koodiaapisen MOOC -kurssille. Loput opettajista ( $n = 32$ ) olivat osallistuneet molempiin mainituista täydennyskoulutuksista. Lisäksi tutkittavia opettajia yhdisti se, että kaikki toimivat opettajina ainakin vuosiluokilla 3–6, jolloin matematiikan opetussuunnitelmassa mukaan astuu graafinen ohjelmointiympäristö. Tällöin kouluopetuksessa ohjelmoinnin saralla ei enää riitä pelkät ilman laitetta tehtävät harjoitukset.

## 5.3 Aineiston analyysi

Saadakseni kattavat vastaukset tutkimuskysymyksiini, päädyin tekemään kuvailvan monimenetelmäisen tutkimuksen, eli hyödynsin niin laadullisia kuin määrällisiäkin tutkimusmenetelmiä.

Aloitin aineistolähtöisen sisällönanalyysin lukemalla vastauksia läpi ilman reuna-merkintöjä ja alleviivauksia. Tällä tavalla pyrin saamaan aineistosta jonkinlaisen kuvan ennen varsinaisen analyysin aloittamista. Seuraavaksi etenin tutkimalla yhden kyselylomakkeessa kysytyn kysymyksen vastauksia – aina yksi kysymys kerrallaan. Etenin aineistolähtöisen analyysimallin mukaisesti aloittaen alkupe-  
räisten vastausten pelkistämällä, jatkaen alaluokkien muodostamiseen ja taas kokoamalla alaluokista relevantteja yläluokkia (Sarajärvi & Tuomi, 2018, 109). Kävin siis mainitut vaiheet jokaisen kyselylomakkeen kysymyksen kohdalla, joista esitän esimerkin taulukossa 2.

Taulukko 2: Esimerkki vastauksen pelkistämisestä sekä ylä- ja alaluokkien muodostamisesta.

<b>Alkuperäinen vastaus</b>	<i>”Olen saanut tietoa, taitoa ja uskallusta. Ei tämä ole vaikeaa, eikä ydinfysiikkaa.” (LO28)</i>			
<b>Pelkistetty ilmaus</b>	Olen saanut tietoa, taitoa ja uskallusta. Ohjelmoinnin opettaminen ei olekaan vaikeaa.			
<b>Alaluokat</b>	Tiedon lisääntyminen	Taidon lisääntyminen	Uskalluksen lisääntyminen	Opettaminen ei olekaan vaikeaa
<b>Yläluokat</b>	Tiedot ja taidot		Oman tekemisen helpottuminen	

Taulukossa 2 on erään opettajan vastaus kysymykseen: Millä tavalla koulutus on vaikuttanut ohjelmoinnin opetukseen? Kuten taulukosta voi nähdä, esimerkkitauksesta löytyi yhteensä neljä alaluokkaa, jotka yhdistyivät kahdeksi yläluokaksi. Tämän kyselylomakkeen kysymyksen kohdalla monen opettajan yksittäisestä vastauksesta löytyi piirteitä eri alaluokista, jotka taas yhdistyivät eri yläluokiksi. Näin ollen yhden opettajan vastaus ei ollut tulkittavissa yhden yläluokan alapuolelle. Kirjoittaessani yhteenvetoa tästä analyysin vaiheesta, koin järkeväksi luokitella vastaukset ensin positiivisiin ja negatiivisiin vastauksiin (ks. kuvio 10, 46), jonka jälkeen etenin ylä- ja alaluokkien esittelemiseen (ks. kuvio 11, 47).

Aineistolähtöisen sisällönanalyysin lisäksi hyödynsin myös määrällistä tutkimusmenetelmää. Ensimmäinen opettajien vastaamista kyselyistä sisälsi yhden likertasteikollisen kysymyksen koskien ohjelmoinnin opetuksen määrää ennen täydennyskoulutusta. Tätä varten laskin kunkin vastauksen antaneiden opettajien määriä ja rakensin tutkimustulokseni tämän jaottelun mukaisesti. Lisäksi tutkimuksestani tekee osin määrällisen myös se, että aineistolähtöisen sisällönanalyysin tehtyäni laskin saamani analyysin perusteella, kuinka monta kertaa opettajat mainitsivat mitään asioita ja lisäsin myös nämä tiedot taulukoiden ja tekstin muodossa tutkimustuloksiini.

Seuraavassa luvussa esittelen tämän tutkimuksen tutkimustuloksia.

## 6 Tutkimustulokset

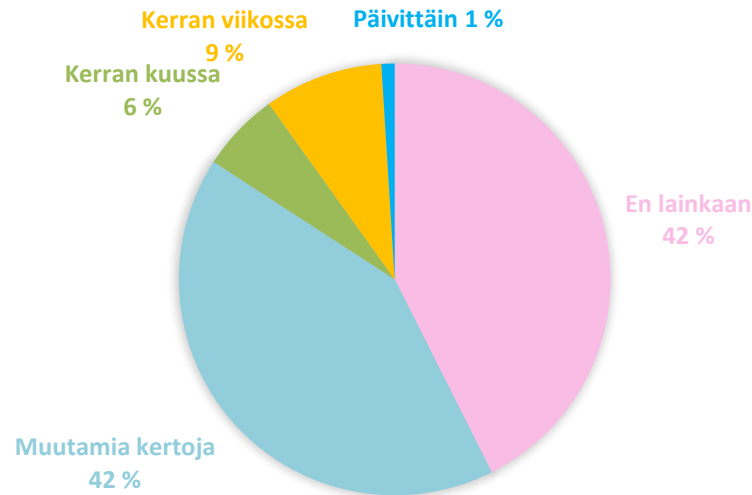
Tässä luvussa esittelen tutkimukseni tuloksia. Alaluvussa *6.1 Täydennyskoulutuksen vaikutukset ohjelmoinnin opettamiseen* tarkastelen ensimmäiseen tutkimuskysymykseen saamiani tuloksia, eli sitä millä tavalla luokanopettajat ajattelevat opettavansa ohjelmointia täydennyskoulutusta ennen, sen jälkeen sekä tulevaisuudessa. Alaluvussa *6.2 Luokanopettajien ajatuksia täydennyskoulutuksen vaikutuksista ohjelmoinnin opettamiseen* puolestaan sisältää tutkimustulokset koskien toista tutkimuskysymystäni.

### 6.1 Täydennyskoulutuksen vaikutukset ohjelmoinnin opettamiseen

Tämän alaluvun olen jakanut ensimmäisen tutkimuskysymyksen mukaisesti kolmeen: Millä tavalla luokanopettajat ajattelevat opettaneensa ohjelmointia ennen täydennyskoulutusta (6.1.1), millä tavalla luokanopettajat ajattelevat opettavansa ohjelmointia täydennyskoulutuksen jälkeen (6.1.2) sekä millä tavalla luokanopettajat ajattelevat opettavansa ohjelmointia tulevaisuudessa (6.1.3). Edellä mainituista jokaisen lopussa on kokoava kaavio, johon on nostettu sellaiset ohjelmoinnin opetuksen toteutustavat, jotka esiintyivät koko aineistossa kaikkein eniten. Tästä eteenpäin käyttäessäni termiä opettaja, tarkoitan tutkimuskohteenani olevia luokanopettajia.

#### 6.1.1 Täydennyskoulutusta ennen

Ensimmäisessä kyselyssä kartoitettiin opettajien ohjelmoinnin opettamisen määrää ennen täydennyskoulutukseen osallistumista (kuvio 3).



Kuvio 3: Vastaajien ( $N = 101$ ) arviot siitä, kuinka paljon he olivat opettaneet ohjelmointia ennen osallistumista täydennyskoulutukseen.

Vastaajista 42 % ( $n = 43$ ) ilmoittivat, etteivät olleet opettaneet ohjelmointia lainkaan. Lähes sama määrä ( $n = 42$ ) opettajia kertoi opettaneensa ohjelmointia muutamia kertoja. Vain 6 % ( $n = 6$ ) ilmoitti opettavansa ohjelmointia kerran kuussa. Tätä suurempi joukko, 9 % ( $n = 9$ ), vastasi opettavansa ohjelmointia kerran viikossa. Yksi vastaaja (1 %) arvioi opettavansa ohjelmointia päivittäin:

*Jos ohjelmoinniksi lasketaan tuntistruktuurin kirjoittaminen joka tunti taululle, teen sitä jatkuvasti. (LO101)*

Ohjelmoinnin opettamisen määrittely ei olekaan aivan yksiselitteistä (ks. Selby & Woollard, 2014; Liukas & Mykkänen, 2014; Futschek, 2006). Tavallaan tuntistruktuurin taululle kirjoittaminen on eräänlainen algoritmien muodostama jono tai resepti (Hyvönen, Lappalainen & Lakanen, 2013; Liukas & Mykkänen, 2014), jota noudattamalla opettajan voi ajatella ikään kuin ohjelmoivansa oppilaita ja oppilaiden lukevan algoritmeja. Mutta nähdäkseni oppilaiden ajattelemisen ohjelmoitavina koneina ei ole laskettavissa ohjelmoinnin opettamiseksi, sillä tämä vaatisi tietoisempaa toimintaa.

### **Kerran viikossa ohjelmointia opettaneet**

Kerran viikossa ohjelmointia opettaneita opettajia oli vastaajista 9 % ( $n = 9$ ). He kaikki mainitsivat jonkin tabletilla tai tietokoneella käytettävän ohjelmointiympäristön. Näissä vastauksissa molemmat sekä Code.org että Scratch, Scratch Junior tai molemmat nousivat esille viidesti. Muita digitaalisia ohjelmointiympäristöjä mainittiin kahdesti. Tästä joukosta neljä kertoi toteuttaneensa ohjelmointi- tai koodauskerhotoimintaa. Kerhotoiminta voi usein olla viikoittaista. Ei olekaan siis ihme, että juuri tästä kerran viikossa ohjelmointia opettavasta joukosta noin puolet sen mainitsivat. Myös robotit näkyivät noin puolissa ( $n = 4$ ) tämän joukon vastauksista. Vain tämän joukon vastauksissa esiin nostettiin ohjelmointikieli ROBO-LAB:

*ROBOLAB on ollut käytössä teknisen käsityön 5.–6.-luokkalaisten oppilaiden kanssa yhteensä kahdeksan oppitunnin jakson ajan. Olen opettanut tällä alustalla vuodesta 1996. (LO2)*

Samalla vastaaja tuli maininneeksi, että on opettanut ohjelmointia jo viime vuosituhannella. Koko aineistosta ( $N = 101$ ) vain viisi vastaajaa kertoi opettaneensa ohjelmointia jo yli kymmenen vuotta sitten. Viikoittain ohjelmointia ennen täydennyskoulutusta opettaneiden ryhmässä nostettiin seuraavat esille kerran: Guru-toiminta, toiminnallisuus sekä ajattelu ohjelmoinnin takana.

### **Kerran kuussa ohjelmointia opettaneet**

6 % ( $n = 6$ ) vastanneista kertoi opettaneensa ohjelmointia ennen täydennyskoulutusta kerran kuussa. Heistä viisi mainitsi jonkin laitteella käytettävän ohjelmointiympäristön. Näissä vastauksissa code.org nousi esiin kolmesti ja Scratch neljästi. Yksi vastaajista (LO9) kertoi käyttäneensä oppilaidensa kanssa Scratchia englanniksi. Kuukausittain ohjelmointia opettaneista kaksi kertoi toteuttaneensa ohjelmoinnin opetusta robotin avulla. Käytetyt robotit olivat Lego-robotti ja Bee-Bot-robotti. Kuukausittain opettaneista opettajista yksi kertoi hyödyntäneensä leikkiä osana ohjelmoinnin opetusta.

## Muutamia kertoja ohjelmointia opettaneet

Opettajista 42 % ( $n = 42$ ) kertoi opettaneensa ohjelmointia muutamia kertoja. Nämä vastaukset sisälsivät yhteensä 32 mainintaa jostain digitaalisesta ohjelmointiympäristöstä. Useimmin mainittiin Code.org, yhteensä 19 kertaa. Yksi näin vastanneista opettajista (LO1) kertoi myös osallistuneensa luokkansa kanssa Hour of Code -tapahtumaan. ESIMERKKI Eräs opettaja taas mainitsi saaneensa ohjelmoinnin opetukseen apua oppilaan vanhemmalta:

*Aloitettu ekaluokan keväällä oppilaan isän ohjauksella. Hän kyseli koodauksesta ja tuli pitämään meille tunnin. (LO4)*

Digitaalisista ohjelmointiympäristöistä esiin nostettiin myös Scratch tai Scratch Junior yhteensä yhdeksän kertaa ja muita ohjelmointiympäristöjä myös yhteensä yhdeksän kertaa. Ilman laitetta ohjelmointia kertoi opettaneen 12 opettajaa. Näistä puolet ( $n = 6$ ) eivät maininneet lisäksi mitään digitaalista ohjelmointiympäristöä, mutta neljä opettajaa nosti ohjelmoinnin toteuttamistavasta kysyttäessä esille loogisen päättelyn:

*Loogisilla paloilla ja esim. Da Vinci koodi -lautapelin avulla harjoiteltu loogista päättelyä. (LO68)*

*Loogista päättelyä, asioiden jakamista osasuorituksiksi, käskyjen järjestämistä oikeaan järjestykseen. (LO82)*

Esimerkeistä jälkimmäisen voisi tulkita sisältävän Liukkaan ja Mykkäsen (2014) algoritmisen ajattelun osa-alueista kaksi: ongelman purkamisen osiin sekä algoritmien luomisen. Ilman laitetta opettaneiden vastauksissa esiintyi seuraavia harjoituksia: opettajan tai luokkakaverin ohjaaminen, kuvan piirtäminen tai palikoiden rakentaminen sekä kirjallisten tai suullisten ohjeiden antaminen ja suorittaminen. Muutamia kertoja opettaneiden vastaukset sisälsivät jonkin robotin:

*Olen opettanut alakouluikäisille ja opettajille Lego-robottien ohjelmointia 1990-luvun loppupuolella. (LO30)*



Myös neljä muuta mainitsi jonkin Lego-robotin. Bee-bot-robottia kertoi käyttäneensä kaksi opettajaa. Yhteistyötä kollegan kanssa kertoi tehneensä viisi muutamia kertoja ohjelmointia ennen täydennyskoulutusta opettaneista opettajista. Heistä neljä kertoi opettaneensa toisille opettajille ohjelmointia. Yksi kertoi rakentaneensa robotin yhdessä oppilaiden ja yhden kollegan kanssa:

*Kolmosten ja osaavan kollegan kanssa rakentamalla Spiker EV3 Lego -robotin. (LO69)*

Erään opettajan (LO37) vastauksesta kävi ilmi, että hän olikin tietämättään opettanut ohjelmointia:

*Vuosikausia olen opettanut koodamista tietämättä, että sitä kutsutaan koodaamiseksi. -- Esim. Loogisia palikoita eri tavoin, palikoiden rakentelua toisen ohjeen mukaan, kuvan piirtämistä ohjeen mukaan, liikkumista ohjeen mukaan jne. (LO37)*

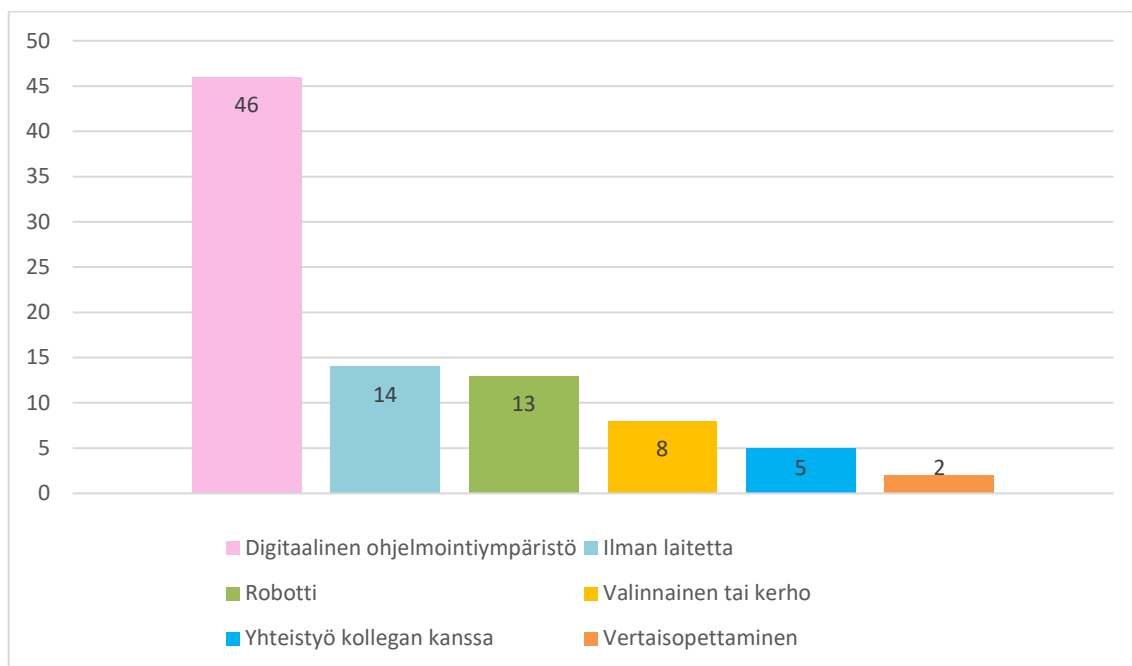
Vastaajien joukosta yksi kertoi pitäneensä ohjelmointikerhoa jo 15 vuotta sitten ja eräs toinen 8.–9.-luokkalaisten valinnaista tietotekniikkaa yli kymmenen vuotta sitten. Valinnainen opetus tai kerhomuotoinen toiminta esiintyivätkin muutamia kertoja opettaneiden joukossa yhteensä neljä kertaa. Samaisesta joukosta kaksi oli keksinyt mainita myös oppiaineen, jonka yhteydessä ohjelmointia oli opettanut – yksi matematiikassa ja toinen käsitöissä. Mainintoja tukee opetussuunnitelma, jossa kyseiset oppiaineet ovat ainoat, joiden sisältöihin kuuluu ohjelmoinnin opetusta (POPS 2014). Muutamia kertoja opettaneista eräs nosti esille myös kummioppilaiden hyödyntämisen:

*Oppilaani ovat siirtäneet osaamistaan kummioppilailleen yhteisillä tunneilla. (LO64)*

Yksi vastaajista ei täsmentänyt, millä tavoin on toteuttanut ohjelmoinnin opetusta aikaisemmin.

## Yhteenveto

Kaikista vastaajista ennen täydennyskoulutusta vähintään muutamia kertoja ohjelmointia opettaneita opettajia oli 58 % ( $n = 58$ ). Kokoon seuraavaksi yhteen, miten nämä opettajat kertoivat toteuttaneensa ohjelmoinnin opetusta ennen täydennyskoulutusta (kuvio 4).



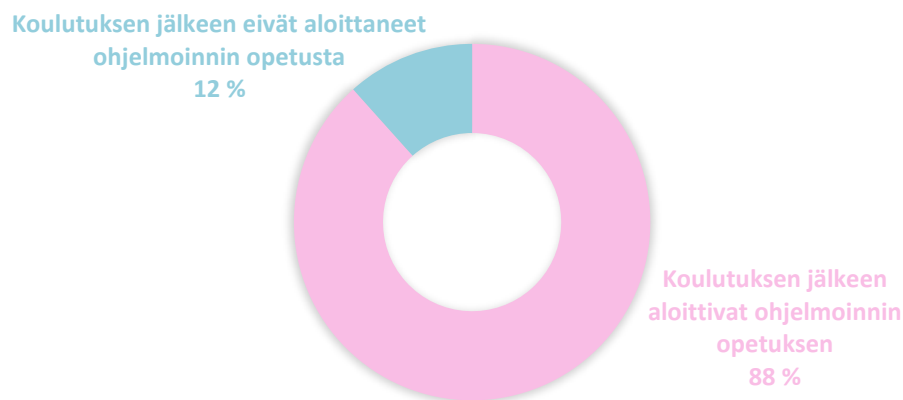
Kuvio 4: Yhteenveto kaikkien ennen täydennyskoulutusta ohjelmointia opettaneiden vastauksista.

Kuten kuviosta 4 voi nähdä, ohjelmointia jo ennen täydennyskoulutusta opettaneiden opettajien vastauksista eniten esille nousi jokin tabletti- tai tietokonekäyttöinen ohjelmointiympäristö. Lähes yhtä usein mainittiin jokin robotti kuin toiminnallinen, ilman laitetta tehtävä harjoitus. Ohjelmointi osana valinnaista opetusta tai kerhotoimintaa nousi esille yhteensä kahdeksan kertaa. Kaksi vastaajaa kertoi hyödyntäneensä oppilaita opettajina. Toisessa tapauksessa isommat oppilaat olivat opettaneet kummioppilailleen ohjelmointia ja toisessa oli kyseessä Guru-toiminta, joka on eräässä kaupungissa käytetty eräänlainen digitutor-oppilas-malli. Opettajista yksi ei kertonut miten oli toteuttanut ohjelmoinnin opetusta aikaisemmin.

Seuraavaksi esittelen, millä tavalla opettajat kertoivat opettaneensa ohjelmointia täydennyskoulutuksen jälkeen.

### 6.1.2 Täydennyskoulutuksen jälkeen

Tässä alaluvussa esittelen, miten opettajat kertoivat opettaneensa ohjelmointia koulutuksen jälkeen. Aloitan tulkitsemalla vastauksia niiltä opettajilta, jotka eivät olleet opettaneet ohjelmointia ennen täydennyskoulutusta. Tämän jälkeen esittelen vastauksia heiltä opettajilta, jotka olivat näin tehneet. Opettajista, jotka eivät olleet opettaneet ohjelmointia lainkaan ennen täydennyskoulutusta, suurin osa kertoi aloittaneensa ohjelmoinnin opettamisen koulutuksen jälkeen (kuvio 5).



Kuvio 5: Ei lainkaan ennen täydennyskoulutusta ohjelmointia opettaneiden vastaukset koulutuksen jälkeen toteuttamastaan opetuksesta.

Ei lainkaan ennen täydennyskoulutusta ohjelmointia opettaneiden vastaukset kielivät täydennyskoulutuksen hyödyllisyydestä, sillä 88 % ( $n = 38$ ) heistä oli aloittanut ohjelmoinnin opettamisen koulutuksen jälkeen. Lopuista 12 prosentista ( $n = 5$ ) neljä eivät vastanneet kysymykseen ohjelmoinnin opettamisesta koulutuksen jälkeen mitään. Tyhjä vastaus on hieman tulkinnanvarainen, sillä se ei väistämättä tarkoita, ettei opetusta olisi aloitettu. Yksi vastaajista kirjoitti, ettei ole toteuttanut ohjelmoinnin opetusta koulutuksen jälkeen mitenkään.

#### Ohjelmoinnin opettamisen koulutuksen jälkeen aloittaneet

Koulutuksen jälkeen ensimmäistä kertaa ohjelmoinnin opettamisen aloittaneiden opettajien ( $n = 38$ ) vastauksissa mainittiin erilaisia digitaalisia ohjelmointiympäristöjä 36 kertaa. Code.org mainittiin yhteensä 15 kertaa:

*Vanhempainillassa katsoimme Code.org-esittelyvideon. (LO94)*

Eräs opettaja (LO94) kertoi tutustuttaneensa koodausta myös oppilaiden vanhemmille. Eräs toinen vastaaja taas kertoi oppilaiden harjoitelleen koodauksen alkeita siten, ettei alkuun antanut ohjeita juuri ollenkaan:

*Olemme harjoitelleet koodauksen alkeita tietokoneilla Code.org:in klassisella sokkelo-ohjelmalla. En tarkoituksella antanut aluksi juurikaan ohjeita vaan kiersin auttamassa, kun joku ei päässyt eteenpäin tai käytti liikaa lohkoja.* (LO100)

Scratch tai Scratch Junior esiintyi ohjelmoinnin opettamisen aloittaneiden vastauksissa useammin kuin Code.org, peräti 18 kertaa. Eräs vastaaja kertoi käyttäneensä Scratchia oppilaiden kanssa tutor-kerhotunneilla, joilla oppilaita koulutettiin, jotta he voisivat opettaa toisia oppilaita. Scratchin käyttäminen S2-oppilaiden kanssa nostettiin esille kerran:

*Omien S2-oppilaitteni kanssa olen tehnyt lyhyitä tarinoita Scratch Junior -sovelluksella.* (LO46)

Muita digitaalisia ohjelmointiympäristöjä kuin Scratchia tai Code.orgia mainittiin yhteensä kolme erilaista. Täydennyskoulutuksen jälkeen aloittaneista yhdeksän kertoi toteuttaneensa opetusta jotain robottia hyödyntäen. Neljässä näistä tapauksissa kyseessä oli Bee-Bot-robotti, kolmessa jokin Lego-robotti ja kahdessa jokin muu. Opettajista kuusi kertoi lähteneensä oppilaiden kanssa liikkeelle siitä, mitä ohjelmointi oikeastaan on:

*Pääsääntöisesti olemme käyneet läpi, mitä ohjelmointi on.* (LO42)

Ohjelmoinnin perusideaa opettajat kertoivat lähestyneensä käytännön ongelmien kautta, kädestä pitäen harjoitellen sekä antamalla oppilaiden itsenäisesti tutustua koodauksen alkeisiin. Yhteensä kuusi ohjelmoinnin opettamisen täydennyskoulutuksen jälkeen aloittaneista opettajista vastasi toteuttaneensa ohjelmointia paperilla koodaamalla. Näissä vastauksissa esiin nousi omien ja valmiiden kuvien

koodaamisen lisäksi myös muun muassa värityskoodausta. Vastauksien joukosta löytyi viisi opettajaa, jotka olivat aloittaneet ohjelmoinnin opettamisen yksinkertaisten, toisten tehtäväksi annettujen, komentojen avulla. Tämän työtavan yhteydessä opettajat mainitsivat niin Multilink-rakennelmat kuin ajattelun taitojen harjoittelunkin. Täydennyskoulutuksen jälkeen aloittaneista opettajista neljän vastauksissa esiintyi jonkinlainen yhteistyö kollegan kanssa:

*Tällä hetkellä on menossa vasta kollegoiden innostaminen aiheeseen.*  
(LO73)

Edellä kuvatun kollegoiden innostamisen lisäksi yhdessä vastauksessa opettaja kertoi tutustuttaneensa muita opettajia aiheeseen. Eräs opettaja vastasi seuraavasti:

*Pitänyt koodauskoulutuksia opettajille ja opettanut ohjelmointia opettajan apuna luokassa (mentorointi), koodikerhon pitäminen 3.–4.-luokkalaisille oppilaille opettajakollegan kanssa.* (LO61)

Edellisen lisäksi myös kaksi muuta opettajaa olivat aloittaneet kerhotoiminnan täydennyskoulutuksen jälkeen. Ohjelmointiin liittyvät ei-digitaaliset pelit puolestaan esiintyivät vastauksissa kolmesti:

*Robogem-lautapeli ja itse tekemäni ohjelmointipeli kartongilla.* (LO86)

*Laivanupotus.* (LO76)

Eräs opettaja (LO17) mainitsi kummioppilaiden hyödyntämisen opetuksen toteuttamisessa:

*Pidimme ohjelmointijakson syksyllä 5. luokan kanssa: Bee-Bot, Scratch Junior ja Scratch. Sen jälkeen oppilaat opettivat ekaluokan kummioppilailleen koodaamista Bee-Botin ja Scratch Juniorin avulla.* (LO17)

Oppilaiden toisilleen opettaminen esiintyi myös vastauksessa, jossa opettaja kertoi järjestäneensä oppilailleen ohjelmointitutor-koulutusta. Lisäksi ohjelmoinnin opettamisen vasta koulutuksen jälkeen aloittaneiden opettajien vastauksissa nousi esille loogisen ajattelun taidot (kahdesti), ohjelmointiprojektipäivän järjestäminen, arkipäivän tilanteiden koodaaminen sekä lyhyiden opetustuokioiden pitäminen.

### **Ohjelmointia jo ennen koulutusta opettaneet**

Opettajat, jotka olivat opettaneet ohjelmointia jo ennen täydennyskoulutusta ( $n = 58$ ), mainitsivat vastauksissaan sekä aiemmin mainitsemiansa että uusia opetusmenetelmiä, materiaaleja ja ajatuksia opettamisensa takana. Samoja opetusmenetelmiä, materiaaleja ja ajatuksia opettamisensa takana kuin ennen täydennyskoulutusta nousi esille 44 kertaa ja uusia, ei aikaisemmin käytettyjä löytyi vastauksista 69 kertaa. Yhden opettajan vastauksessa saattoi ilmetä useita eri menetelmiä, materiaaleja tai ajatuksia. Seuraavaksi esittelen tarkemmin, mitä nämä vastaukset pitivät sisällään.

Digitaaliset ohjelmointiympäristöt olivat koulutuksen jälkeen suosittuja myös ohjelmoinnin opettamisen jo ennen täydennyskoulutusta aloittaneiden joukossa. Niin samojen kuin uusienkin ohjelmien käyttöä esiintyi useissa vastauksissa. Eräs opettaja kertoi näin:

*Lähinnä matematiikan tunneilla. Jakanut Code.org-sivun oppilaille myös kotona tehtäväksi. (LO38)*

Edellinen vastaus oli poikkeuksellinen, sillä kyseinen opettaja oli hyödyntänyt Code.org:ia jo ennen täydennyskoulutusta, mutta täydennyskoulutuksen jälkeen ryhtynyt käyttämään ohjelmointiympäristöä uudella tavalla. Ilman laitetta käytettävien ei-aikaisemmin-käytettyjen ohjelmointiympäristöjen ja -harjoituksien määrä lisääntyi:

*Olemme 4.-luokkalaisten kanssa tehneet joitain robottitehtäviä niin että pareittain ohjaavat toisiaan, lukeneet Linda Liukkaan Hello Ruby -kirjan, tutustuneet Koodaustuntiin ja Scratch Junioriin. (LO21)*

Myös yksi toinen opettaja mainitsi ottaneensa Hello Ruby -kirjan osaksi ohjelmoinnin opetustaan. Tällaisista ei-digitaalisista ympäristöistä ja tehtävistä esille nousi nyt myös ensimmäistä kertaa esimerkiksi matematiikan oppikirjan koodustehtävät. Robottien kanssa toimimista jatkettiin, mutta sitä myös aloitettiin:

*Olen pitänyt Bee-Bot-pajoja, ja kannustin kouluamme tilaamaan Bee-Bot-robotteja. (LO63)*

Edellä kuvattu opettaja olikin ainoa, joka tässä vaiheessa kertoi olleensa edesauttavana henkilönä koulun hankkiessa uusia laitteita. Bee-Bot-robottien lisäksi myös Lego-robottien kanssa jatkettiin. Joukossa oli myös opettajia, jotka aloittivat Lego-robottien kanssa toimimisen ensimmäistä kertaa täydennyskoulutuksen jälkeen. Samalla tavalla kävi myös oppilaiden toistensa opettamisen kohdalla: yksi opettaja raportoi jatkaneensa ja kolme opettajaa kertoi aloittaneensa tällaisen toiminnan:

*Olemme kutosten kanssa tehneet oman pienen animaation Scratchilla ja kutoiset ovat pelanneet Code.org -pelejä kakkosten kanssa ja opettaneet kakkosia. (LO60)*

Kukaan vastaajista ei maininnut jatkaneensa yhteistyötä kollegojen kanssa, mutta yhteensä kahdesta vastauksesta ilmeni, että tämänkaltaista toimintaa oli aloitettu koulutuksen jälkeen. Kerhotoimintaa taas kertoi jatkaneensa yhteensä kaksi opettajaa ja aloittaneensa yksi opettaja. Kaikkiaan kolmesta vastauksesta ilmeni suoraan, että täydennyskoulutuksen antia oli hyödynnetty opetuksessa. Ensimmäinen opettaja oli koostanut täydennyskoulutuksen (Kiinnostaako koodaus) materiaaleista oman harjoitusmonisteen, toinen kertoi hyödyntäneensä täydennyskoulutuksen (Koodiaapisen MOOC) rakennetta omassa opetuksessaan ja kolmas taas mainitsi koulutuksessa saadut ideat ylipäänsä. Eräs opettaja kuvaili lähteneensä liikkeelle suunnitelmallisesti:

*Olemme lähteneet suunnitelmallisesti liikkeelle askel askeleelta, esim. lauseke: mietimme ensin mitä se oikein tarkoittaa leikkien avulla tai [mietimme] esimerkkejä arjesta, missä niitä käytetään ja sitten siirrymme käyttämään lauseketta esim. Scratch-ohjelmassa. (LO96)*

Lisäksi vastauksissa oli nähtävillä erilaisia yksittäisiä näkökulmia, kuten käskyjen merkitys, ajattelun taidot, arjen esimerkit opetuksessa, koko koulun koodiviikko sekä oman luokan ohjelmointiprojektit. Eräs opettaja raportoi, että koulutuksen jälkeen aika oli kulunut toistaiseksi vain tulevan suunnitteluun. Hän ei siis ollut opettanut ohjelmointia koulutuksen jälkeen. Yksi opettajista kertoi hyödyntäneensä oman koulunsa nimettyä verkkopedagogia:

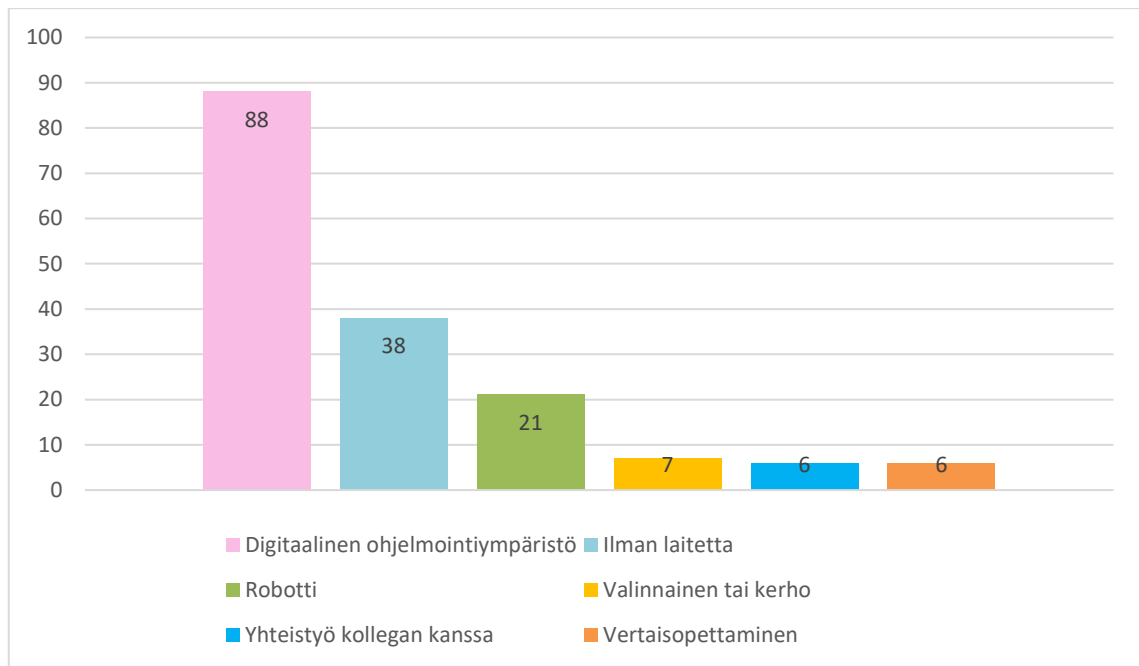
*Opettamalla sitä itse sekä verkkopedon avustuksella pääasiassa tablettyöskentelynä. (LO97)*

Edellä olevan vastaus oli ainoa koko aineistossa, jossa tällainen verkkopedon hyödyntäminen mainittiin. Tämä saattaa johtua siitä, että kyseinen toimintamalli (tällä nimellä) on tiettävästi käytössä vain yhdessä suomalaisessa kaupungissa.

## **Yhteenveto**

Kokoan seuraavaksi yhteen, miten kaikki kyselyyn vastanneet opettajat kertoivat toteuttaneensa ohjelmoinnin opetusta täydennyskoulutuksen jälkeen (kuvio 6).





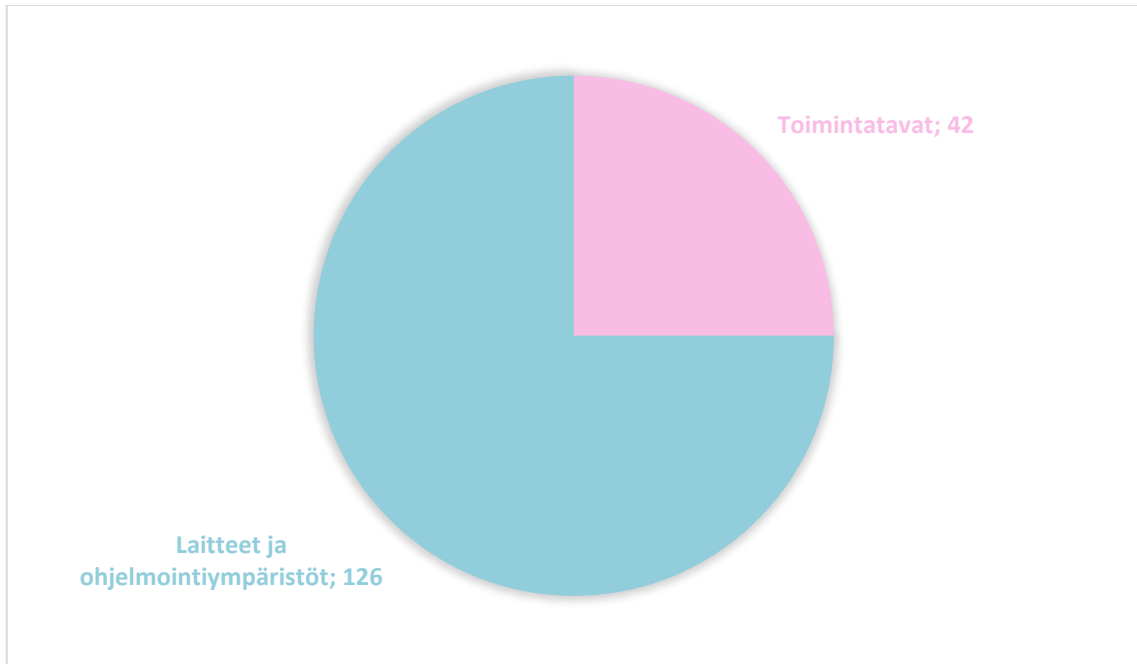
Kuvio 6: Yhteenveto kaikkien opettajien vastauksista koskien täydennyskoulutuksen jälkeen toteutettua ohjelmoinnin opetusta.

Kuvio 6 osoittaa, että digitaalisia ohjelmointiympäristöjä käytettiin ylivoimaisesti eniten täydennyskoulutuksen jälkeen: Erilaisia digitaalisia alustoja mainittiin vastauksissa yhteensä 88 kertaa. Toiseksi eniten vastauksissa esiintyivät erilaiset ilman laitetta tehtävät harjoitukset ja pelit, yhteensä 38 kertaa. Erilaiset robotit mainittiin 21 kertaa ja valinnainen opetus tai kerhotoiminta kaikkiaan seitsemän kertaa. Oppilaiden vertaisopettaminen sekä kollegan kanssa toteutettu yhteistyö nousivat aineistosta esille yhtä monta kertaa, molemmat mainittiin kuudesti.

Siirryn seuraavaksi tarkastelemaan sitä, miten opettajat vastasivat aikeistaan opettaa ohjelmointia tulvaisuudessa.

### 6.1.3 Tulvaisuudessa

Kaikista vastaajista 93 vastasi lomakkeen kysymykseen koskien tulevaisuuden aikeita opettaa ohjelmointia. Seuraavaksi esittelen näitä tuloksia (kuvio 7).

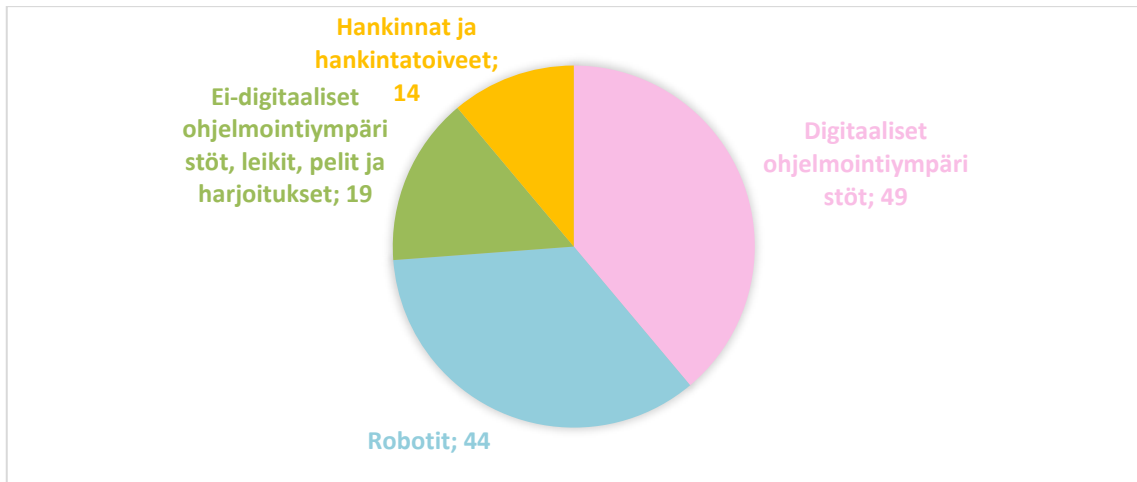


Kuvio 7: Maininnat koskien opettajien vastauksia ohjelmoinnin opettamisesta tulevaisuudessa.

Taulukosta 7 näkee, että opettajien vastaukset koskien ohjelmoinnin opettamista tulevaisuudessa, sisälsivät kaikkiaan 126 laitteita ja ohjelmointiympäristöjä koskevaa mainintaa. Erilaisia toimintatapoja taas mainittiin yhteensä 42 kertaa. Seuraavaksi esittelen laitteita ja ohjelmointiympäristöjä sisältäneitä vastauksia tarkemmin, jonka jälkeen toimin samoin toimintatapoja sisältäneiden vastausten suhteen.

### **Laitteet ja ohjelmointiympäristöt**

Laitteita ja ohjelmointiympäristöjä koskevat maininnat olen jakanut neljään (kuvio 8).



Kuvio 8: Laitteita ja ohjelmointiympäristöjä koskevat maininnat opettajien vastauksissa koskien ohjelmoinnin opettamista tulevaisuudessa.

Kuviosta 8 näkee, että kaikkiaan vastauksissa ilmeni erilaisia digitaalisia ohjelmointiympäristöjä yhteensä 49 kertaa. Suurin osa (30) näistä maininnoista koski Scratch tai Scratch Juniorin käyttöä. Ohjelman avulla haluttiin muun muassa syventää oppilaiden taitoja. Erilaisista selainpohjaisista alustoista useimmin mainittiin Code.org sekä Kodu. Ohjelmointiympäristöt, leikit, pelit ja harjoitukset, joiden käyttäminen ei vaadi teknologiaa nostettiin esille 19 kertaa. Näissä vastauksissa esiintyi useasti loogista ajattelua sekä ajattelun taitoja ylipäänsä edistäviä tehtäviä:

*Koitan miettiä monipuolisia tapoja opettaa sitä myös ilman teknisiä välineitä, koska niitä ei ole ainakaan nyt remontin väistötiloissa toimiessa helppo saada luokkaan. (LO62)*

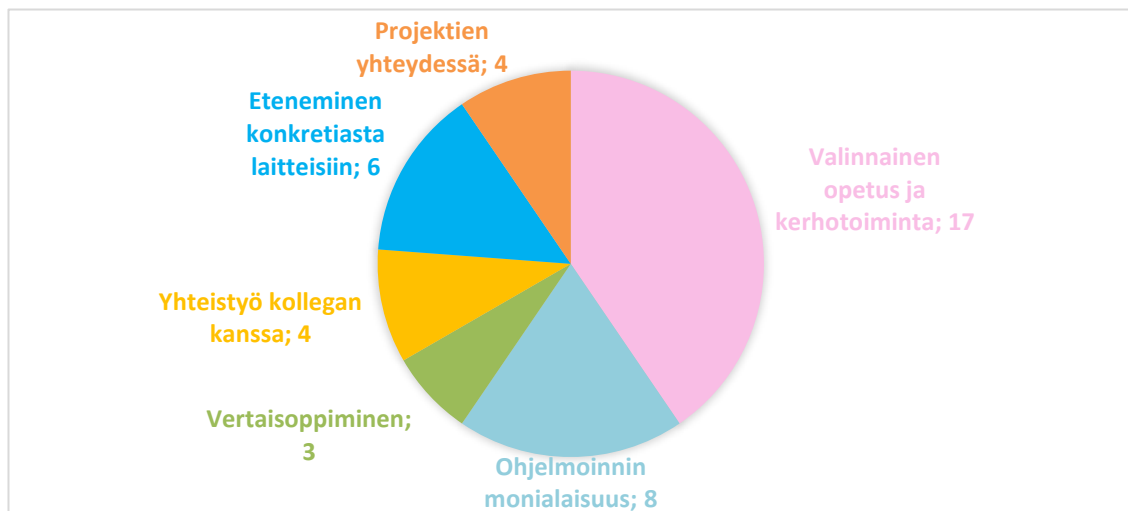
Yllä olevan kaltaisia vastauksia, joissa resurssit johtuisivat väistötiloihin joutumisesta ei noussut esille muita. Resurssit nostettiin esille useammin hankintoja kohdalla. Opettajista kolme raportoi jatkavansa robotiikan opettamista:

*Lisäksi jatkan oman luokkani vitosten kanssa pieninä rupeamina robo-tunteja, koska aloitimme robo-projektin jo kolmannella eTwinning-projektissa, vaikka en tuolloin robotiikasta yhtään mitään tiennyt tai osannut. (LO69)*

Robottiikkaa kerrottiin jatkettavan esimerkiksi teknisen käsityön tunneilla. Suurin osa opettajista, jotka raportoivat toteuttavansa ohjelmoinnin opetusta tulevaisuudessa robottien avulla, mainitsivat Lego-robotit, Bee-Botit tai Blue-Botit. Useat halusivat myös hankkia robotteja koululle, jos niitä ei vielä ollut. Opettajat kertoivat myös erilaisista hankinnoista sekä hankintatoiveistaan. Eräs vastaaja kertoi haluavansa hankkia jonkun laitteen. Toinen opettaja ilmoitti, että hänen koululleen oli tulossa iPadit. Erilaisista Lego-roboteista, Bee-Boteista ja Blue-Boteista haaveili kaikkiaan kuusi opettajaa. Neljä opettajaa kertoi hankkineensa Lego-robotteja ja kaksi opettajaa jotain muita robotteja. Ei-digitaalisia hankintoja oltiin tehty myös, sillä eräs opettaja kirjoitti hankkineensa koululleen erilaisia logiikkapelejä, kuten Loogiset palat -pelin.

### Toimintatavat

Toimintatapoja koskevat vastaukset koostuvat kuudesta eri luokasta (kuvio 9).



Kuvio 9: Toimintatapoja koskevat maininnat opettajien vastauksissa koskien ohjelmoinnin opettamista tulevaisuudessa.

Kuviosta 9 näkee, että valinnainen opetus tai kerhotoimintaa esiintyi opettajien vastauksissa 17 kertaa. Vastauksissa toistuivat koodauskerhot ja robotiikkakerhot. Lisäksi erikseen mainittiin ”Looginen ajattelu, koodaus ja robotiikka” -kurssi, Scratch-kerho sekä tieto- ja viestintäteknologian valinnaisen kurssin ohjelmointiteema. Eräs opettaja kertoi uskaltavansa kirjoittamaan koulunsa opetussuunnitelmaan Robo-kurssin:

*Vaikka omat taidot superheikot, uskaltauduin kirjoittamaan uuteen OPSiin vapaavalintaisiin valintoihin Robo-kurssin ja ainakin ensi syksyksi oppilaita on sen valinnut ja luultavimmin päädyn sitä yhtenä opettajana vetämään, tuntiresursseista riippuen. (LO69)*

Opettajat kertoivat myös ottavansa ohjelmoinnin osaksi kaikkea opetusta tai toteuttavansa sitä oppiainerajat ylittävästi kahdeksassa vastauksessa:

*Ensi vuonna haluaisin yhdistää ohjelmoinnillista ajattelua ja visuaalista koodausta tekstitaitojen opiskeluun (sekä kirjoittamiseen että kirjallisuudenopiskeluun). (LO46)*

*Ohjelmointi tulee olemaan osa jopa kaikkea opetusta kuten äidinkielessä, uskonnossa. (LO2)*

Äidinkielen ja uskonnon lisäksi ohjelmointi aiottiin ottaa osaksi matematiikkaa, ympäristöoppia, historiaa, taiteen tekemistä sekä laajempia monialaisia oppimiskokonaisuuksia. Tulevaisuutta suunnitellessa opettajista vain kaksi kertoi aikovansa hyödyntää vertaisoppimista muun muassa tutor-oppilas-mallin avulla. Kollegiaalinen yhteistyö nousi esille neljästi:

*Olla muidenkin opettajien kanssa ideoimassa/toteuttamassa ohjelmoinninopetuksen käynnistämistä eri luokka-asteilla koulussamme. (LO14)*

Lisäksi samanaikaisopettajuus mainittiin kerran ja eräs opettajista kertoi lähtevänsä Tansaniaan kouluttamaan paikallisia opettajia Bee-Botien sekä Scratchin saloihin. Kollegiaalisen yhteistyön lisäksi yksi opettaja paljasti toiveensa viritellä yhteistyötä paikallisen ohjelmointiyrityksen kanssa. Eräs opettaja (LO48) kertoi edistäneensä koko koulun toimintakulttuuria ohjelmoinnin ja koodauksen saralla:

*Kirjoitan juuri koulullemme ohjelmoinnin/koodauksen ohjelmaa osaksi tvt-strategiaa ja siinä olen samalla luonut tulevalle lukuvuodelle omaan opetukseen tavoitteita. (LO48)*

Muutamissa vastauksissa nousi esille halu opettaa ohjelmointia viikoittain. Suunnittelun korostuminen näkyi kahdessa vastauksessa:

*Kaikille pitää korostaa suunnittelua: Ensin mietitään, mitä ajotaan tehdä, sitten suunnitellaan ja sitten toteutetaan. (LO43)*

*Tärkeintä sitoa käytännön tilanteisiin ajattelun taitojen kehitys; aikataulutus, tehtävien pilkkominen osiin, huolellinen suunnittelu jne. (LO23)*

Ajattelun taidot tuntuivat korostuvat vastauksissa, joissa ilmeni, että opettaja oli aloittamassa työskentelyn alkuopetuksen parissa:

*Aloitan ykkösluokalla ensi lukuvuonna ja aion opettaa ensin ajattelun taitoja siten, että oppilaat neuvovat toisilleen pareittain piirtämistä, 3-ulotteisten rakennelmien kokoamista ja reitin etsimistä paperille piirrettyssä sokkelossa. (LO100)*

*Siirryn 1.luokan opeksi, joten ajattelun taidot korostuvat. Aion opettaa niitä viikoittain säännöllisesti yhdistäen eri oppiaineisiin. (LO71)*

Lisäksi aineistosta nousi esille esimerkiksi pedagogisen ajattelun soveltaminen, järjestelmällisemmin opettaminen, mahdollisimman monipuolisesti opettaminen, jonkin uuden opettaminen, koodauksen opettaminen välipaloina, harjoittelun jatkaminen sekä opetuksen määrän lisääminen. Yksi opettaja nosti esille myös ohjelmoinnin opettamisen kontekstisidonnaisuuden kirjoittamalla, että tulevaisuudessa ohjelmoinnin toteuttamisen tavat riippuvat oppilaista ja heidän tarpeistaan.

Kaksi vastaajaa ei osannut sanoa, miten aikoo opettaa ohjelmointia tulevaisuudessa:

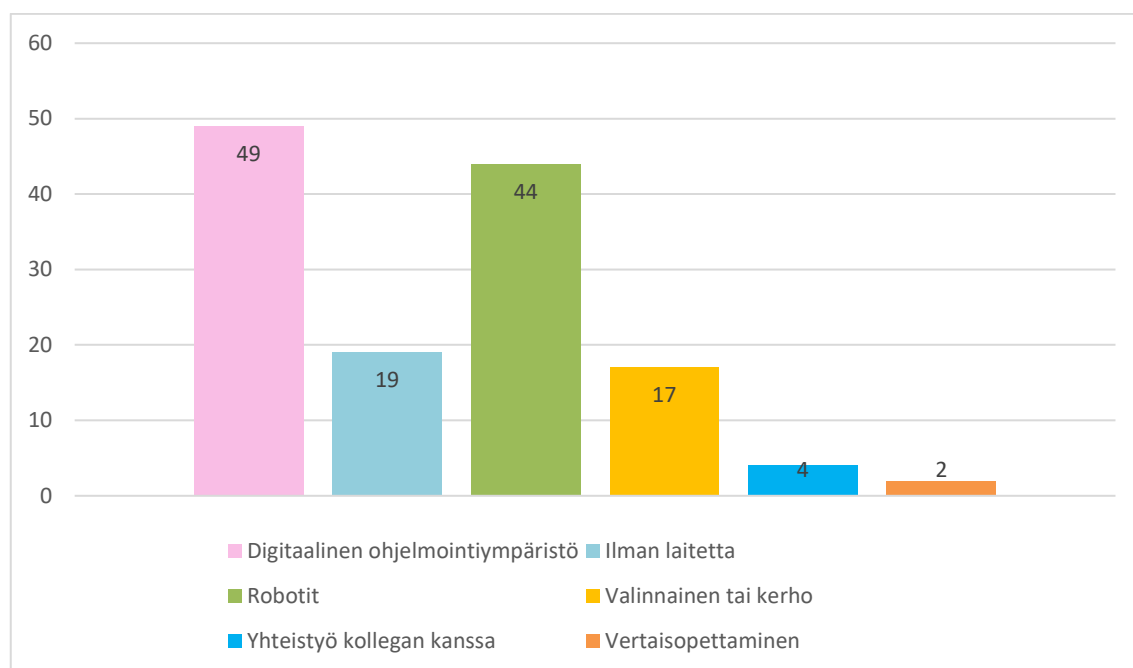
*En osaa sanoa, kun en itse osaa mitään. (LO53)*

Tällainen vastaus, jossa kerrottiin, ettei opettaja itse osaa mitään, oli todella harvinainen, eikä sellaisia esiintynyt muita tulevaisuuden opettamisen toteutuksesta

kysyttäessä. Yksi toinen vastasi myös, ettei osaa sanoa tulevasta, mutta hänen syynään oli tuleva työpaikan vaihdos.

## Yhteenveto

Kuten aiemmissakin alaluvuissa, kokoon tässäkin yhteen opettajien vastauksista nousseita tiettyjä opetuksen toteuttamisen tapoja lukumäärittäin (kuvio 10).



Kuvio 10: Yhteenveto kaikkien opettajien vastauksista koskien tulevaisuudessa toteutettavaa ohjelmoinnin opetusta.

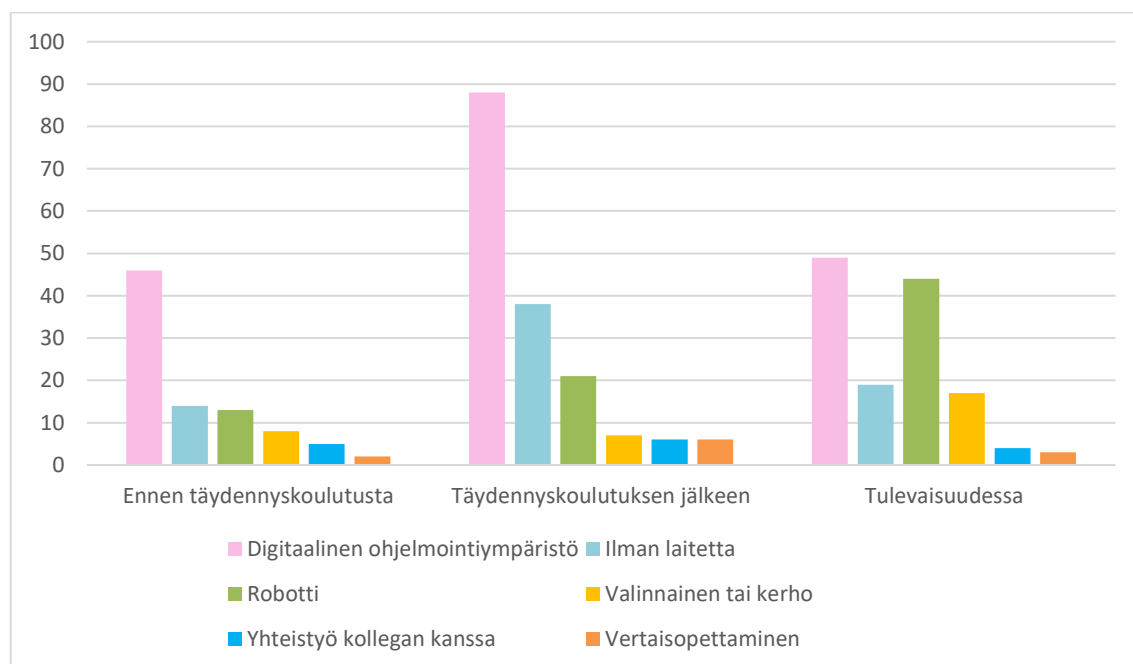
Kuvio 10 osoittaa, ettei digitaalisia ohjelmointiympäristöjä aiottu käyttää ylivoimaisesti eniten tulevaisuudessa, sillä myös erilaiset robotit nostettiin esille lähes yhtä monta kertaa. Samoin valinnaista opetusta tai kerhotoimintaa koskevia vastauksia nousi esille lähes yhtä monta kertaa kuin ilman laitetta tapahtuva ohjelmointiakin. Neljä opettajaa vastasi aikovansa toteuttaa yhteistyötä kollegan kanssa ja kaksi opettajaa raportoi aikovansa hyödyntää vertaisoppimista.

Seuraavaksi kokoon yhteen tämän alaluvun antia koskien täydennyskoulutuksen vaikutuksia.

### 6.1.4 Yhteenveto

Tässä alaluvussa esitän kokoavasti edellisten alalukujen antia. Opettajien vastaukset koskien heidän toteuttamaansa ohjelmoinnin opetusta vaihtelivat riippuen

siitä, kartoitettiin tilannetta ennen täydennyskoulutusta, täydennyskoulutuksen jälkeen vai tulevaisuudessa (kuvio 11).



Kuvio 11: Opettajien ajatukset ohjelmoinnin opettamisesta ennen täydennyskoulutusta, sen jälkeen sekä tulevaisuudessa.

Kuvion 11 avulla voimme tulkita opettajien ohjelmoinnin opetuksen toteuttamisen muutoksia. Jokaisesta kolmesta vaiheesta (ennen, jälkeen sekä tulevaisuudessa) opettajat nostivat esille tiettyjä opetuksen toteuttamisen tapoja, joista kuusi esiintyi kaikissa vaiheissa. Nämä kuusi olivat digitaalisen ohjelmointiympäristön käyttäminen, ilman laitetta työskentely, robottien hyödyntäminen, valinnaisen opetuksen tai kerhotoiminnan järjestäminen, yhteistyö kollegan kanssa sekä oppilaiden vertaisopettaminen.

Digitaalisia ohjelmointiympäristöjä käytettiin jo ennen koulutusta huomattavasti enemmän kuin muita työskentelytapoja. Täydennyskoulutuksen jälkeen opettajien vastauksista nousi esille kaksinkertainen määrä digitaalisten ohjelmointiympäristöjen hyödyntämistä. Opettajien tulevaisuudensuunnitelmissa digitaalisten ohjelmointiympäristöjen käyttö mainittiin enää suunnilleen yhtä monta kertaa kuin ennen täydennyskoulutusta. Tämä on erityisen mielenkiintoista siksi, että ennen täydennyskoulutusta ylipäänsä ohjelmointia opettaneiden joukko koostui vain 58



opettajasta kun tulevaisuudessa tapahtuvaan ohjelmoinnin toteuttamista koskevaan kysymys koski koko joukkoa ( $N = 101$ ).

Ohjelmoinnin toteuttaminen ilman digitaalisia laitteita mukaili digitaalisten ohjelmointiympäristöjen trendiä: Alun alkaen ennen koulutusta tällaista laitteetonta opetusta raportoi hyödyntävänsä noin kolmannes digitaalisten ohjelmointiympäristöjen käyttäjien määrästä. Kuten digitaalisten ympäristöjen kohdalla, myös laitteettomia harjoituksia kerrottiin hyödynnettävän kaksinkertaisesti täydennyskoulutuksen jälkeen verrattuna ennen tätä. Tulevaisuudensuunnitelmia tarkastellessa laitteettomien harjoitusten määrä puolittui aikaisempaan nähden.

Sen sijaan robotteja hyödyntävien opettajien määrä kasvoi jokaisella mittauksella. Täydennyskoulutuksen jälkeen robotteja hyödyntävien opettajien määrä kasvoi kolmanneksella verrattuna koulutusta edeltäneeseen aikaan. Tämän jälkeen lukumäärä peräti kaksinkertaistui tulevaisuuden ohjelmoinnin opettamisen toteuttamisesta kysyttäessä.

Valinnaista ohjelmointiin liittyvää opetusta tai kerhotoimintaa raportoitiin harjoitettavan lähes yhtä paljon ennen ja jälkeen täydennyskoulutuksen. Opettajia, jotka suunnittelivat toteuttavansa tulevaisuudessa tämäntyyppistä toimintaa, oli lopulta suunnilleen kaksinkertainen määrä alkutilanteeseen ja täydennyskoulutuksen jälkeiseen aikaan nähden.

Yhteistyöstä kollegan kanssa raportoineiden opettajien määrä ei juuri vaihdellut tarkasteltavien ajanjaksojen välillä. Lukumäärä oli alun alkaenkin pieni ( $n = 5$ ). Täydennyskoulutuksen jälkeen yhteistyötä toisten opettajien kanssa tekevien opettajien määrä nousi vain yhdellä, ja tulevaisuuden näkymiä tarkastellessa aihe nousi esiin enää vain neljän opettajan vastauksissa.

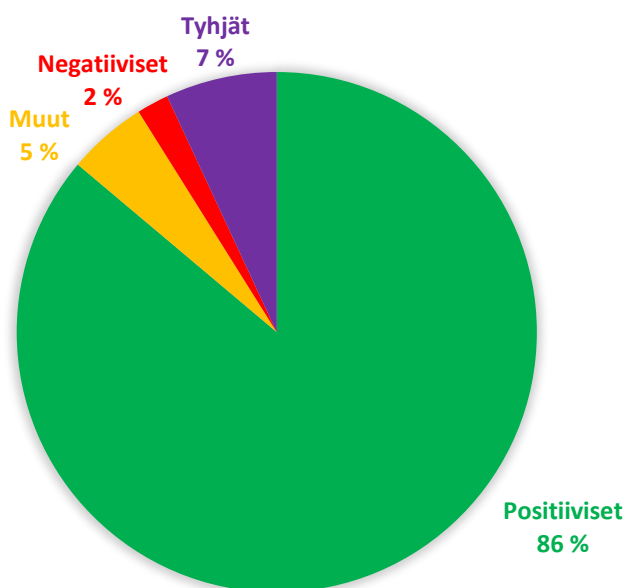
Oppilaiden vertaisopettamisesta raportoi aluksi vain kaksi opettajaa. Täydennyskoulutuksen jälkeen tällaista ohjelmoinnin opettamisen toteuttamistapaa kirjoitti hyödyntävänsä kolminkertainen ( $n = 6$ ) määrä opettajia. Lopulta tulevaisuuden suunnitelmissa aihe nousi esille enää vain kolmesti.

Ohjelmoinnin opettamisen toteuttamisesta tulevaisuudessa kysyttäessä esille nousi myös sellaisia teemoja, joita ei ollut ollenkaan tai oli vain hyvin vähän näkyvillä ennen täydennyskoulutusta tai täydennyskoulutuksen jälkeen. Nämä esille nousseet teemat olivat hankinnat, projektit, monialaisuus sekä opetuksen eteneminen ilman laitetta tapahtuvasta opetuksesta laitteiden käyttöä vaativiin ohjelmoinnin harjoituksiin. Sitä, etteivät nämä teemat nousseet esille heti täydennyskoulutuksen jälkeen, selittää aika: kyselylomakkeisiin vastattiin niin pian täydennyskoulutuksen jälkeen, ettei opettajat olleet ehkä ehtineet vielä toteuttaa opetusta täysin haluamallaan tavalla.

Seuraavassa alaluvussa esittelen opettajien ajatuksia täydennyskoulutuksen vaikutuksista ohjelmoinnin opettamiseen.

## 6.2 Luokanopettajien ajatuksia täydennyskoulutuksen vaikutuksista ohjelmoinnin opettamiseen

Täydennyskoulutuksen jälkeen opettajilta myös kysyttiin, millä tavalla koulutus on vaikuttanut heidän ohjelmoinnin opetukseensa. Vastaukset heijastelivat opettajien tärkeinä pitämiään asioita. Näistä vastauksista valtaosa oli tulkittavissa positiivisiksi (kuvio 12).

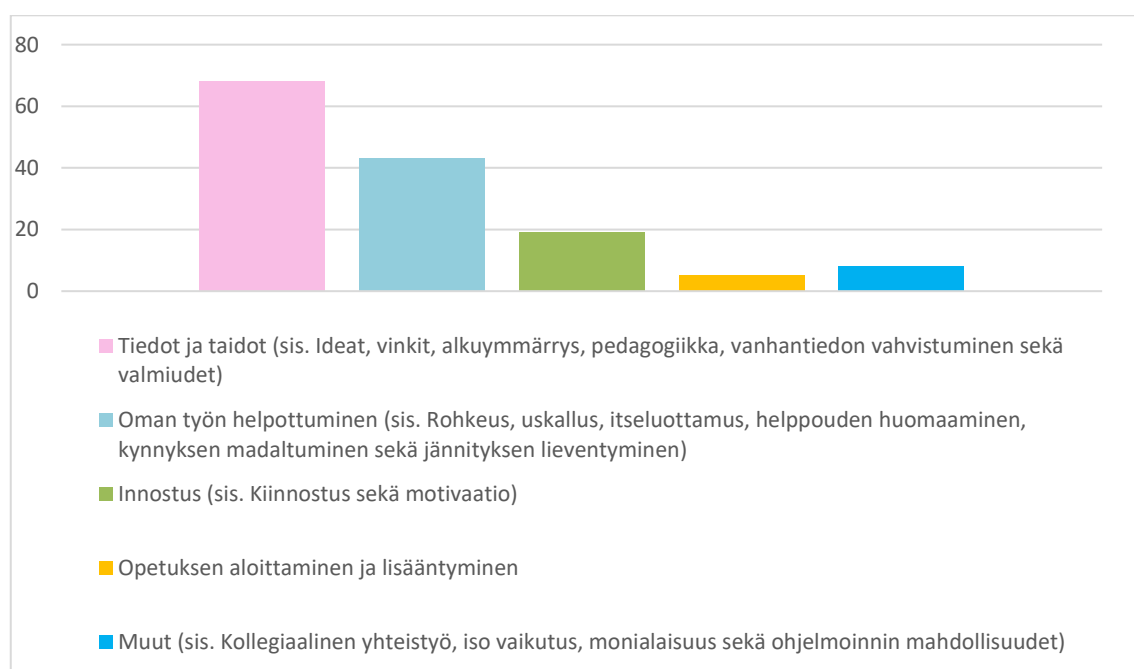


Kuvio 12: Opettajien vastaukset siitä, millä tavalla koulutus vaikutti heidän ohjelmoinnin opetukseensa jaoteltuina positiivisiin, negatiivisiin, tyhjiin sekä muihin vastauksiin.

Kuviosta 12 käy ilmi, että peräti 86 % ( $n = 87$ ) kaikista vastauksista oli tulkittavissa pelkästään positiivisiksi. Vain negatiiviksi tulkittavia vastauksia oli 2 % ( $n = 2$ ) ja muita vastauksia 5 % ( $n = 5$ ). Kaikkiaan seitsemän opettajaa (7 %) olivat jättäneet vastaamatta kysymyksen. Seuraavaksi pureudun näihin vastauksiin tarkemmin aloittaen positiivisista vastauksista.

## Positiiviset

Positiivisiksi luokiteltujen vastausten ( $n = 87$ ) sisältä oli löydettävissä kaikkiaan viisi eri luokkaa, jotka sisältävät erilaisia teemoja (kuvio 13).



Kuvio 13: Täydennyskoulutuksen positiiviset vaikutukset luokittain.

Kuvio 13 auttaa hahmottamaan, minkälaisia teemoja opettajat nostivat esille koken täydennyskoulutuksen vaikutuksia heidän ohjelmoinnin opetukseensa. Opettajilta kysyttiin, millä tavalla koulutus on vaikuttanut heidän ohjelmoinnin opetukseensa.

### Tiedot ja taidot

Kaikkein useimmin vastauksissa nousivat esille erilaiset tiedot ja taidot. Tietojen karttuminen nousi vastauksissa esille yhteensä 14 kertaa. Tietojen karttumisen yhteydessä opettajat eivät useinkaan määritelleet mitä tietoja he olivat saaneet.

Muutamissa vastauksissa tiedot saivat kuitenkin selityksiä, kuten käsitteiden oppiminen ja tieto erilaisista opetusmenetelmistä. Ohjelmointiin liittyvien taitojen lisääntyminen nousi opettajien vastauksissa esille kymmenesti. Taitojen yhteydessä mainittiin esimerkiksi käytännön taidot, perustaidot sekä uudet taidot. Eräs opettaja vastasi opettamisen mahdollistuneen taitojen karttumisen myötä:

*Ope on saanut itse taitoja, jotta voi opettaa! (LO91)*

Mainintoja ideoista ja vinkeistä löytyi opettajien vastauksista yhteensä 24:

*Hyviä ideoita, joita ensi vuonna pääsee hyödyntämään. (LO73)*

*Olen saanut varmuutta ja taitoja omaan ohjelmointiin, sekä hyviä ideoita ohjelmoinnin opettamiseen. (LO3)*

Ideoihin ja vinkkeihin liittyvät maininnat esiintyivät vastauksissa useimmiten ilman sen suurempaa selitettä kuin ”ideoita ohjelmoinnin opettamiseen”, mutta poikkeuksiakin löytyi. Eräs opettaja raportoi saaneensa vinkkejä nimenomaan käytävistä ohjelmista, toinen taas mainitsi ideoiden antaneen paljon ajateltavaa ja kehittäväää. Vastauksissa esiin nousseita alkuymmärrykseen liittyviä teemoja oli kaikkiaan seitsemän:

*Sain ideoita ja "aloituspaketin". (LO7)*

*Sain siitä alkuymmärryksen siitä, miten ohjelmointia opetetaan. (LO9)*

Pedagogiikka nousi esille viidesti. Siitä raportoitiin esimerkiksi näin:

*Mietin tarkemmin oppilaiden osaamistaitoja ja minkälaisia harjoituksia olisi hyvä toteuttaa, kuinka selkeästi esitettävä. (LO61)*

*Osaan suunnitella tunnit rakenteellisesti paremmin ja ymmärrän ohjelmoinnin pedagogiikan paljon paremmin. (LO96)*

Kaikkiaan viidessä vastauksessa oli tulkittavissa vanhan tiedon vahvistumista. Opettajista kaksi ilmaisi oman osaamisen vahvistuneen. Eräs opettaja ilmaisi myös vanhan tiedon muistuneen taas mieleen:

*Koulutus on palauttanut mieleeni monta vanhaa asiaa, joita voin edelleen käyttää työssäni. (LO37)*

Valmiuksien lisääntyminen nostettiin esille kolmesti: valmiudet ylipäänsä, valmiudet ohjelmoinnin opettamiseen sekä valmiudet kokeilla ohjelmointia oppilaiden kanssa eri ympäristöissä.

#### Oman työn helpottuminen

Opettajat nostivat esille myös erilaisia asioita, jotka ovat helpottaneet oman opetustyötä. Tähän luokkaan kuuluvan rohkeuden tai uskalluksen opettajat nostivat esille 20 kertaa:

*Se on antanut rohkeutta lisää viedä asia luokkahuoneeseen ja siitä on saanut ideoita. (LO88)*

*Olen uskaltanut aloittaa ohjelmoinnin opetuksen. (LO55)*

*Antanut rohkeutta ohjelmoinnin opettamiseen. (LO49)*

Edellisten lisäksi rohkeuden raportoitiin vaikuttaneen esimerkiksi tuntemattomienkin ohjelmointialustojen käyttämiseen. Myös itseluottamus, itsevarmuus sekä usko omaan osaamiseen näkyi aineistossa 14 kertaa.

*Antanut varmuutta ja uskoa omaan osaamiseen. (LO46)*

Eräs opettaja myös raportoi, että täydennyskoulutus vaikutti hänen opetukseensa luomalla uskoa, että hän itse voi oppia. Opettajista kolmen vastauksissa oli nähtävillä yhteensä neljästi se, ettei ohjelmointi ole vaikeaa. Eräs opettaja perusteli vaikeuden olemattomuutta uskalluksella lähteä kokeilemaan. Toinen opettaja taas vertasi ohjelmointia ydinfysiikkaan:

*Ei tämä ole vaikeaa, eikä ydinfysiikkaa. (LO28)*

Kaikkiaan kolme opettajaa ilmaisivat koulutuksen vaikuttaneen heidän ohjelmointin opettamiseen riman madaltumisena:

*Madalsi kynnystä aloittaa ohjelmoinnin opettaminen. (LO17)*

Jännityksen ja paniikin hälventymisestä opettajat kirjoittivat seuraavasti:

*Kiinnostus koodausta kohtaan on itsellä lisääntynyt, ei jännitä tehdä luokan kanssa asioita. (LO39)*

*Ei niin paniikkia ensi lukuvuotta ajatellen. (LO101)*

### Innostus

Innostus tai into nousi opettajien vastauksissa esille 16 kertaa. Innostuksen raportoitiin vaikuttaneen niin ohjelmoinnin opettamisen aloittamiseen kuin robotiikan opettamisen jatkamiseenkin. Monissa innostumiseen liittyvissä vastauksissa innostaminen mainittiin kuitenkin vain sellaisenaan:

*Innostanut. (LO81)*

Edellä mainitun lisäksi kahdessa vastauksessa nostettiin esille kiinnostuksen lisääntyminen ohjelmointia kohtaan. Eräs opettaja kirjoitti myös hänen motivaationsa nousseen. Lisäksi yksi opettaja raportoi koulutuksen olleen silmiä avaava ohjelmoinnin opetusmahdollisuuksien suhteen.

### Opetuksen aloittaminen ja lisääntyminen

Kaikkiaan kaksi opettajaa ilmaisivat opetuksensa lisääntyneen täydennyskoulutuksen jälkeen. Opettajista kolme raportoi aloittaneensa opetuksen koulutuksen jälkeen:

*Kokeilin ohjelmointia vasta koulutuksen jälkeen, sen innostamana. (LO89)*

### Muut

Opettajat nostivat esille myös muita, vaikeasti luokiteltavia teemoja. Vastauksissa esiintyi esimerkiksi muiden opettajien vaikutukset sekä heidän kanssaan tehty yhteistyö:

*Verkostoiduin muiden koulujen opettajien kanssa. (LO80)*

*Uusia näkökulmia ja vinkkejä vertaisilta eli muilta opettajilta. (LO24)*

Toiset kaksi opettajaa taas mainitsivat koulutuksen vaikuttaneen ylipäänsä merkittävästi:

*Nollasta lähdettiin... Joten paljon. (LO33)*

Ohjelmoinnin monialainen hyödyntäminen sekä sen liittäminen monenlaiseen toimintaan mainittiin opettajien vastauksissa kahdesti:

*Ohjelmointiajattelun saa sisällytettyä monenlaiseen toimintaan koulussa eikä ohjelmointi oikeastaan ole vaikeaa, kunhan vaan uskaltaa lähteä kokeilemaan. (LO76)*

### **Negatiiviset**

Negatiiviseksi tulkittavia koulutuksen vaikutuksiin liittyviä vastauksia oli kaikkiaan kaksi kappaletta, eli vain 2 % kaikista tutkimukseen valituista opettajista:

*Koulutus on ahdistanut, kun ei osaa. (LO1)*

*Huomasin, että se on vaikeaa, itse en oppinut mitään, koska kouluttajat eivät neuvoneet minua. (LO53)*

**Muut**

Vastauksia, joiden ei voitu luotettavasti tulkita kuuluvan positiivisiin tai negatiivisiin, oli kaikkiaan viisi. Eräs opettaja vastasi koulutuksen vahvistaneen omia ajatuksia. Vastauksesta ei kuitenkaan käynyt ilmi, mitä nämä omat ajatukset olivat. Kaksi opettajaa taas vastasivat, etteivät olleet vielä löytäneet aikaa ohjelmoinnin opettamiselle. Yhdessä vastauksessa ilmaistiin, ettei koulutus vaikuttanut juurikaan.

*Vahvistanut omia ajatuksia. (LO11)*

*Ei juurikaan. (LO84)*

Seuraavaksi tarkastelen tämän tutkimuksen luotettavuutta.



## 7 Luotettavuus

Arvioin tutkimukseni luotettavuutta seuraavien elementtien kautta: tutkimuksen kohde ja tarkoitus, oma sitoutumiseni tutkijana, aineiston keruu, tutkimuksen ajankohta ja kesto, aineiston analyysi sekä tutkimuksen raportointi (Sarajärvi & Tuomi, 2018, 140–141).

Tutkimuksessani tutkin täydennyskoulutuksen vaikutuksia luokanopettajien ohjelmoinnin opetukseen. Tutkimuskohteenani olleet täydennyskoulutuksen käyneet luokanopettajat tukivat siis tutkimuksen tarkoitusta. Huolellisesti laadittu tutkimuksen teoria koskien aikaisempia täydennyskoulutuksen vaikutuksia lisäsi tämän tutkimuksen luotettavuutta. Näin siksi, että kytkökset aiemman tutkimuksen ja tämän tutkimuksen välillä lisäsi tutkimukseni yleistettävyyttä (Sarajärvi & Tuomi, 2018).

Myös oma sitoutumiseni tutkijana lisää tutkimukseni luotettavuutta. Huolimatta omista ennakko-oletuksista tutkimustuloksia kohtaan, pyrin toimimaan läpi tutkimusprosessin objektiivisena tutkijana. Näin välttyttiin siltä, että tutkimustuloksiin olisi vahingossa sisällynyt jotain tutkijan mielikuvien värjäämää. On kuitenkin todettava, että tutkimusprosessin tapahtuessa pääosin yksinään työskennellen, vaara objektiivisuuden katoamiselle on olemassa. Näin siksi, että omia päätelmiä yksin pohtiessa ei voi olla aivan varma ihastuuko omaan kädenjälkeensä hieman liian sokeasti vähentäen samalla tutkimuksen luotettavuutta (Sarajärvi & Tuomi, 2018).

Tutkimuksen aineisto kerättiin vuosina 2015–2016. Luotettavuuden lisäämiseksi on hyvä tiedostaa, että tällöin ohjelmointi oli vasta tulossa saman vuoden syksynä nykyiseen perusopetuksen opetussuunnitelmaan. Tämä saattaa vaikuttaa tulosten myönteisyyttä horjuttavasti, sillä iso osa opettajista vastasi, ettei ollut opettanut lainkaan ohjelmointia ennen täydennyskoulutusta. Mikäli aineisto olisi ollut tuoreempi, kerätty aikana, jolloin ohjelmointi oli jo osa opetussuunnitelmaa, täydennyskoulutus ei kenties olisi näyttänyt vaikuttaneen ohjelmoinnin opetuksen aloittamista lisäävästi.

Aineistolähtöisen analyysin perusongelmana nähdään se, että teorialähtöistä analyysiä pidetään yleisesti luotettavampana (Sarajärvi & Tuomi, 2018, 96). Tämän vuoksi tutkimusta toteuttaessa tärkeäksi muodostui Pohdinta-luvussa esitellyn tutkimustulosten kytköksen aiempaan tutkimukseen ja tietoon täydennyskoulutuksen vaikuttavuudesta.

Tutkimuksen raportointiin kohdistuvaa luotettavuutta lisää se, että tutkimuksen toteutusta on pyritty avaamaan mahdollisimman yksityiskohtaisesti (Sarajärvi & Tuomi, 2018; Kyngäs, Elo, Pölkki, Kääriäinen & Kanste, 2011). Näin mahdollistuu tutkimuksen toistettavuus, joka osaltaan lisää tutkimuksen luotettavuutta. Tutkimustuloksia raportoidessani olen pyrkinyt kirjoittamaan auki vain sen, mitä vastaajat ovat itse kirjoittaneet. En myöskään väitä missään kohtaa tutkimustulosteni näyttelevän täydennyskoulutusten saralla kokonaista totuutta. Esimerkiksi aineiston analyysia auki kirjoittaessa oli hyvä tiedostaa, että tyhjät vastaukset ovat tulkinnanvaraisia. Tämän vuoksi pyrin kirjoittamaan tutkimustuloksia mahdollisimman neutraalisti tuoden esille myös tyhjien vastausten määrän.

Seuraavassa luvussa pohdin tutkimukseni keskeisimpiä tuloksia sekä niiden pohjalta syntyneitä uusia näkökulmia.

## 8 Pohdinta

Tässä tutkimuksessa selvisi, että kysyttäessä opettajilta, miten he ovat toteuttaneet ohjelmoinnin opetusta ennen täydennyskoulutusta, sen jälkeen sekä tulevaisuudessa, vastauksista oli löydettävissä yhtäläisyyksiä. Opettajat nostivat esille muun muassa erilaisia ohjelmointiympäristöjä, robotteja, yhteistyön kollegan kanssa, valinnaisen opetuksen tai kerhotoiminnan sekä oppilaiden vertaisoppimisen. Vaikka opetuksen toteuttamistavoissa oli selkeitä yhtäläisyyksiä kaikissa tarkasteluvaiheissa, opetus myös lisääntyi ja monipuolistui täydennyskoulutuksen myötä.

Vuonna 2019 tehdyn valtioneuvoston selvityksen mukaan vain noin 22 % opettajista oli kokeillut oppilaiden kanssa graafista ohjelmointia (Tanhua-Piironen, 2019, 28). Tässä tutkimuksessa kuitenkin selvisi, että täydennyskoulutuksen käyneet opettajat nostivat juuri erilaiset ohjelmointiympäristöt kaikkein useimmiten esille. Esimerkiksi Scratchin ja erilaisten robottien käyttämisen yleistymisen täydennyskoulutuksen myötä voisi ajatella olevan yhteydessä siihen, että Kiinnostaako koodaus -koulutuksessa opettajat osallistuivat työpajoihin, joissa käytiin konkreettisesti läpi tällaisten ohjelmointiympäristöjen ja robottien käyttöä. Code.org-ympäristön suosiota selittää edellisen seikan lisäksi myös sen veloitukseton saatavuus millä tahansa selaimeen pääsevällä laitteella (Code.org, 2019). Erilaisten laitteiden olemassaoloa ja toimivuutta Vuorinen (2019, 106) pitää ehtona ohjelmoinnin opetuksen onnistumiselle. Saman suuntaisia päätelmiä voisi tehdä myös tämän tutkimuksen tuloksista, sillä erilaiset robotit nousivat opettajien vastauksissa esille usein. Ohjelmointiympäristöihin liittyviä mainintoja oli lähes kaksinkertainen määrä koulutuksen jälkeisestä ohjelmoinnin opetuksen toteuttamisesta kysyttäessä. Kun kysyttiin ohjelmoinnin opettamisen aikeista tulevaisuudessa, mainintojen määrät laskivat. Tästä voisi päätellä, että kyseessä on ollut jonkinlainen alkuinnostus tällaisia opetuksen toteuttamisen tapoja kohtaan. Robotteihin liittyvät maininnat sen sijaan kasvoivat läpi tarkasteluvaiheiden.

Kiinnostaako koodaus -koulutuksessa opettajat saivat tietoa erilaisista tavoista toteuttaa yhteistyötä kollegan kanssa. Tämä näkyi myös tutkimustuloksissa, sillä

heti erilaisten ohjelmointiympäristöjen ja robottien hyödyntämisen jälkeen kolmanneksi useimmiten opettajien vastauksissa nousi esille erilaisia tapoja toteuttaa yhteistyötä kollegan kanssa. Tämän voi ajatella hyvänä asiana, sillä esimerkiksi Fisherin, Byrnen ja Tangneyn vuonna 2016 Irlannissa toteuttamassa tutkimuksessa opettajien sosiaaliskonstruktivistinen tiimityöskentelyn malli osoittautui tehokkaaksi. Tehokkuus syntyi vapaan keskustelun mahdollistamista ideoista sekä ryhmän asiantuntijuuden hyödyntämisestä. Myös Suomessa valtioneuvosto on ehdottanut, että koulujen (opettaja)digitutortoimintaa tulisi kehittää, jotta opettajien teknologinen ja pedagoginen osaaminen kohenisi (Tanhua-Piiroinen, 2019, 1). Maininnat kollegiaalisesta yhteistyöstä pysyivät kuitenkin lähes samoissa lukemissa läpi tarkastellun ajanjakson.

Opettajien tulevaisuuden suunnitelmat koskien ohjelmoinnin opettamista sisälsi yli kaksinkertaisen määrän mainintoja oppilaiden vertaisopettamisesta nähden siihen, mitä he raportoivat toteuttaneensa heti täydennyskoulutuksen jälkeen. Valinnaisopetuksella onkin Suomessa pitkät juuret, sillä valinnaista ATK-opetusta annettiin ensimmäisen kerran jo 1960-luvulla (Saarikoski, 2004).

Kuten ohjelmointiympäristöihin liittyvissä maininnoissa, myös vertaisoppimisessa kävi niin, että koulutuksen jälkeen oli havaittavissa jonkunlaista alkuinnostusta, sillä tulevaisuudessa toteutettavasta ohjelmoinnin opetuksesta kysyttäessä vertaisoppiminen ei noussutkaan esille enää yhtä monesti. Vertaisoppimisen merkitystä olisi ehkä siis syytä korostaa koulutuksissa entistä enemmän, sillä oppilaslähtöistä opetusta pidetään tärkeänä ja oppilaiden osallistamista oppilaiden kannalta hyödyllisenä (Norrena, 2015; Taina, 2015).

Kun opettajilta kysyttiin, millä tavalla koulutus oli vaikuttanut heidän ohjelmoinnin opetukseensa, suurin osa kertoi vain positiivisista vaikutuksista. Erityisesti opettajat raportoivat tietojensa ja taitojensa karttumisesta, oman työn tekemisen helpottumisesta sekä innostuksen lisääntymisestä.

Opettajien tietojen ja taitojen karttuminen koulutuksen myötä voidaan nähdä pelkästään positiivisena asiana. Myös Norrena (2015) ja Vuorinen (2019) peräänkuuluttavat pedagogisen näkökulman tärkeyttä teknologian hyödyntämisessä.

Ohjelmoinnin opettamisen kannalta hyvää on se, että opettajien vastauksissa esiintyi mainintoja algoritmisen ajattelun ymmärtämisestä. Näin siksi, että ohjelmoinnin opettaminen voidaan nähdä pohjimmiltaan juuri algoritmiseen ajatteluun sisältyvien taitojen opettamisena (Liukas & Mykkänen, 2014, 78). Opettajien tietojen ja taitojen karttuminen pienentää tasoeroja opettajien ohjelmointitaidoissa, jolloin taataan oppilaille tasa-arvoisemmat lähtökohdat oppia ohjelmointia ja algoritmista ajattelua (Muhonen, Kaarakainen & Savela, 2015).

Opettajat raportoivat oman työnsä helpottumisesta esimerkiksi lisääntyneen rohkeuden ja itsevarmuuden kautta. Opettajien kokema itsevarmuus onkin olennainen osa erilaisten digilaitteiden käyttämistä opetuksessa (Tanhua-Piironen ym., 2019, 29; Siddig & Scherer, 2016).

Tutkimuksessa nousi esille myös opettajien innostuminen ohjelmointia kohtaan täydennyskoulutuksen myötä. Samanlaisia tuloksia sai myös Hietikko kumppaneineen, kun täydennyskoulutuksen käyneet opettajat näkivät digitalisaation innostavana ja suhtautuivat myös omiin kykyihinsä itsevarmemmin (Hietikko ym., 2016, 20).

Suurimmat poikkeamat tutkimuksessani koskivat sellaisia opettajia, jotka kokivat ahdistusta tai epätoivoa siitä, etteivät he osanneet ja ymmärtäneet ohjelmointia. Tällaisia opettajia oli alle 2 % vastaajista ( $N = 101$ ). Myös Vuorinen (2019, 105) nostaa esille täydennyskoulutuksien päämäärien saavuttamisen ongelmallisuuden silloin, kun opettajien teknologinen osaaminen on niin heikkoa, että se rajoittaa itse ohjelmoinnin oppimista.

Tutkimukseni jatkoksi mielenkiintoinen ja ajankohtainen jatkotutkimusaihe olisi tutkia sellaisia opettajia, jotka eivät ole itse käyneet ohjelmoinnin täydennyskoulutusta, mutta ovat saaneet opastusta täydennyskoulutuksen käyneeltä kollegalta. Tällöin voisi verrata koulutuksen itse käyneen ja opastusta koulutuksen käyneeltä kollegalta saaneen ajatuksia työn tekemisen helpottumisesta ja koulutuksen vaikutuksista.

Tutkimukseni tuloksia voi hyödyntää, kun halutaan perustella täydennyskoulutuksen hyötyjä. Lisäksi saamani tulokset voidaan nähdä voimavarana ohjelmoinnin täydennyskoulutusta kehitettäessä. Kaikkiaan tulokset näyttävät positiivisilta jo nyt, olihan osa opettajista jopa uskaltanut aloittaa ohjelmoinnin opetuksen ihan ensimmäistä kertaa vasta täydennyskoulutuksen käytyään.

## Lähteet

- Blikstein, P. (2015). *Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design*. Foundation and Trends in Human-Computer Interaction
- Code.org (2019). *About Us*. Saatavilla: <https://code.org/international/about>
- Erilaisten oppijoiden liitto ry (2019). *Ötökkä tuo ohjelmoinnin perusteita oppitunneille* Saatavilla: <http://lukiapuvaline.fi/apuvalineet/robotit/otokka-ohjelmoinnin-perusteita-oppitunneille/>
- Fisher, L., Byrne, J. R., & Tangney, B. (2016). Teacher Experiences of Learning Computing using a 21st Century Model of Computer Science Continuing Professional Development. IN *CSEDU* (2) (273–280). Saatavilla: <http://www.scitepress.org/Papers/2016/59067/59067.pdf>
- Ford, M. (2015). *Robottien kukoistus. Teknologia ja massatyöttömyyden uhka*. Turku: Kustannusosakeyhtiö Sammakko.
- Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. IN *International conference on informatics in secondary schools- evolution and perspectives* (159–168). Berlin: Springer.
- Halinen, I., Kartovaara, E., Manninen, M., Nyyssölä, K., & Packalen, P. (2009). *Näkökulmia perusopetuksen tavoitteisiin ja tuntijakoon*. Muistio. Opetushallitus.
- Hatlevik, I. K., & Hatlevik, O. E. (2018). Examining the Relationship Between Teachers' ICT Self-Efficacy for Educational Purposes, Collegial Collaboration, Lack of Facilitation and the Use of ICT in Teaching Practice. *Frontiers in psychology*, 9/2018. Saatavilla: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6008425/pdf/fpsyg-09-00935.pdf>
- Hietikko, P., Ilves, V. & Salo, J. (2016). *Askelmerkit digiloikkaan*. OAJ:n julkaisusarja 3:2016.
- Hyvönen, M., Lappalainen, V., & Lakanen, A. J. (2013). Ohjelmointi 1: C. *Luentomoniste/Jyväskylän yliopisto, tietotekniikan laitos*, (17). Jyväskylä: Jyväskylän yliopistopaino. Saatavilla: <https://jyx.jyu.fi/bitstream/handle/123456789/47417/978-951-39-4859-7.pdf?sequence=1>

Innokas Network (2019). *Kumppanit*. Saatavilla: <https://www.innokas.fi/kumppanit/>

Kekäläinen, O. (2015) Onko automatisointiajattelu paras suomennos käsitteestä ”computational thinking”? Teoksessa Viteli, J. & Östman, A. (toim.) *Tuovi 13: Interaktiivinen tekniikka koulutuksessa 2015-konferenssin tutkijatapaamisen artikkelit* (27–29). Tampereen yliopisto.

Korkeakoski, E. (toim.). *Opettajien täydennyskoulutuksen tuloksellisuus*. Opetushallituksen rahoittama pitkäkestoinen koulutus. Arviointi, 3, 1999. Saatavilla: [http://oph.fi/download/49192\\_opettajien\\_taydennyskoulutuksen\\_tuloksellisuus.pdf](http://oph.fi/download/49192_opettajien_taydennyskoulutuksen_tuloksellisuus.pdf)

Kyngäs, H., Elo, S., Pölkki, T., Kääriäinen, M. & Kanste, O. (2011). Sisällönanalyysi suomalaisessa hoitotieteellisessä tutkimuksessa. *Hoitotiede* 2011, 23(2), 138–148.

LEGO Engineering (2019). *Platform: ROBOLAB*. Saatavilla: <http://www.legoengineering.com/platform/robolab/>

LEGO.com (2019). *LEGO Mindstorms EV3. Käyttöopas*.

Lekolar (2019). *Ohjelmointi*. Saatavilla: <https://www.lekolar.fi/koulu/verkkokauppa/opetusvalineet/ohjelmointi/>

Lindfors, E., & Pirttimaa, M. (2018). Teknologiakasvatuksen haasteet ja mahdollisuudet automaatioteknologian oppimisympäristössä. *Ainedidaktikka*, 2(1), 2–19. Saatavilla: <https://journal.fi/ainedidaktikka/article/view/60823/32825>

Mikkola, A. (2012). Preface: Perspectives for the Future of the Teaching Profession. Teoksessa Niemi, H., Toom, A. & Kallioniemi, A. (toim.) *Miracle of Education: The Principles and Practices of Teaching and Learning in Finnish Schools* (ix–xi). Sense Publishers.

Muhonen, M., Kaarakainen, M-T. & Savela, J. (2015). Opettajien teknologiataidot oppilaiden tulevaisuuden taitojen (epä)tasa-arvoisuuden edistäjinä? Teoksessa Viteli, J. & Östman, A. (toim.) *Tuovi 13: Interaktiivinen tekniikka koulutuksessa 2015-konferenssin tutkijatapaamisen artikkelit* (56–64). Tampereen yliopisto.



- Mykkänen, J., & Liukas, L. (2014). Koodi 2016. Helsinki: Lönnberg Print. Saatavilla: [https://s3-eu-west-1.amazonaws.com/koodi2016/Koodi2016\\_LR.pdf](https://s3-eu-west-1.amazonaws.com/koodi2016/Koodi2016_LR.pdf)
- Nelimarkka, M., Vainio, N. & Kinnunen, N. (2011). *Ohjelmointia Scratchin kanssa*.
- Norrena, J. (2015). *Innostava koulun muutos - Opas laaja-alaisen osaamisen opetukseen*. Jyväskylä: Ps-kustannus.
- Paaso, A. (2012). Osaava ammatillinen opettaja 2020. Teoksessa Nokelainen, P. (toim.) *Tutkimus ammatillisen opettajan tulevaisuuden työnkuvasta*. Rovaniemi: Lapin yliopisto.
- POPS (2014). *Perusopetuksen opetussuunnitelman perusteet 2014*. Helsinki: Opetushallitus.
- Quahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences*, 191, 1479–1482.
- Rousku, K. (toim.), Andersson, C., Stenfors, S., Lähteenmäki, I., Limnell, J., Mäkinen, K., Kopponen, A., Kuivalainen, M. & Rissanen, O. P. (2019). *Pilkahduksia tulevaisuuteen. Tietopolitiikka, tekoäly ja robotisaatio hyvinvoinnin ja taloudellisen menestyksen mahdollistajana Suomessa. Valtiovarainministeriön julkaisuja 2019:22*.
- Saarikoski, P. (2004). *Koneen lumo. Mikrotietokoneharrastus Suomessa 1970-luvulta 1990-luvun puoliväliin* (väitöskirja, Turun yliopisto). Jyväskylän nykykulttuurin tutkimuskeskuksen julkaisuja 83. Jyväskylä 2004.
- Saarikoski, P. (2006). Koneen ja koulun ensikohtaaminen. *Tekniikan Waiheita*, 24(3), 5–19.
- Salo, M., Kankaanranta, M., Vähähyyppä, K., & Viik-Kajander, M. (2011). Tulevaisuuden taidot ja osaaminen. Asiantuntijoiden näkemyksiä vuonna 2020 tarvittavasta osaamisesta. Teoksessa Kankaanranta, M. & Vahtivuori-Hänninen, S. (toim.), *Opetusteknologia koulun arjessa II* (19–40). Jyväskylä: Jyväskylän yliopistopaino. Saatavilla: <https://jyx.jyu.fi/bitstream/handle/123456789/37469/1/978-951-39-4616-6.pdf#page=20>
- Sarajärvi, A., & Tuomi, J. (2018). *Laadullinen tutkimus ja sisällönanalyysi*. Tammi.

- Selby, C., & Woollard, J. (2014). *Computational thinking: the developing definition*. Saatavilla: [https://eprints.soton.ac.uk/356481/1/Selby\\_Woollard\\_bg\\_soton\\_eprints.pdf](https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf)
- Siddiq, F., & Scherer, R. (2016). The relation between teachers' emphasis on the development of students' digital information and communication skills and computer self-efficacy: the moderating roles of age and gender. *Large-scale Assessments in Education*, 4(1), 17.
- Standl, B. (2017, November). Solving everyday challenges in a computational way of thinking. IN *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (180–191). Cham: Springer.
- Taina, J. (2015). Peruskoulun ohjelmointiopetus. *Matematiikkalehti Solmu*, 2/2015, 1-2.
- Tanhua-Piiroinen, E., Kaarakainen, S. S., Kaarakainen, M. T., Viteli, J., Syvänen, A., & Kivinen, A. (2019). Digiajan peruskoulu. *Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja*, 6/2019. Saatavilla: [http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161383/6-2019-Digiajan%20peruskoulu .pdf?sequence=1&isAllowed=y](http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161383/6-2019-Digiajan%20peruskoulu.pdf?sequence=1&isAllowed=y) PILKO TÄMÄ VIELÄ OSIIN
- Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. IN *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (120–129). ACM.
- Verkkokauppa.com (2019). *LEGO Mindstorms*. Saatavilla: <https://www.verkkokauppa.com/fi/catalog/11150c/LEGO-Mindstorms>
- Vuorinen, J. (2019). Opettajien täydennyskoulutus, ohjelmointi ja ohjelmoinnillinen ajattelu. Teoksessa Portaankorva-Koivisto, P., Heinonen, M., & Mäkelä, E. (toim.). *Kuka meitä opettaa? Esseitä tietotekniikan opetuksesta* (91–115).
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35. Saatavilla: <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf>